



US Army Corps  
of Engineers

Construction Engineering  
Research Laboratory

AD-A227 946

DTIC FILE COPY

USACERL Technical Report P-90/31, Vol II

September 1990

National Test Facility Consulting

## Scheduling for Cost-Plus Construction Projects, Volume II: PROMAN Source Files

by

E. William East

In cost-plus fixed fee contracts, the contractor is paid based on the actual costs incurred plus some graded incentive fee, rather than on a fixed bid. The larger the project, the more critical it becomes to ensure that the impact of all schedule changes be efficiently analyzed, so that the contractor can complete intermediate milestones as well as the entire project on time. Commercially available scheduling software does perform many of the necessary analyses, but cannot filter data in ways that are useful to schedule reviewers. In planning large projects, there is an acute need for supplemental software that can process scheduling data to make it immediately useful to schedule reviewers. The PROMAN (PROject MANagement) microcomputer-based schedule analysis software was developed by the U.S. Army Construction Engineering Research Laboratory (USACERL) to provide several tools to help evaluate progress of the construction project, to assess the impact of schedule changes upon the project, and to provide other assistance in construction schedule management.

In this study, PROMAN was employed on a large cost-plus project undertaken by the Corps of Engineers in the construction of the National Test Facility (NTF) in Colorado Springs, CO. USACERL provided regular consulting services to NTF, to monitor the contractor's progress schedule and to identify requirements for additional schedule analysis tools. Volume I of this report includes specific recommendations for the National Test Facility Resident Office and future cost-plus projects to assist in evaluating progress schedules and documentation to guide in the installation and use of PROMAN scheduling software. Volume II includes source code for the PROMAN software. The results of this study apply to all cost-plus construction projects and may also be applied to other large Corps construction projects.

2  
DTIC  
ELECTED  
OCT 16 1990  
S E D  
(RP)

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official indorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

***DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED***

***DO NOT RETURN IT TO THE ORIGINATOR***

## Notice to Program Recipients

This program is furnished by the U.S. Government and is accepted and used by the recipient with the express understanding that the Government makes no warranty, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof.

The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program. The recipient also agrees that the program and all documents related thereto, including all copies and versions (except when expressly authorized otherwise) in possession thereof, will be discontinued from use or destroyed upon request by the Government.

The program is to be used only in the public interest and/or the advancement of science and will not be used by the recipient to gain unfair advantage over any client or competitor. Whereas the recipient may charge clients for the ordinary costs of applying the program, the recipient agrees not to levy a charge, royalty, or proprietary usage fee (except to cover any normal copying and/or distribution costs) upon any client for the development or use of the received program. Recipients desiring to modify and remarket the program will be required to comply with a separate agreement. Only minor or temporary modifications will be made to the program (e.g., necessary corrections or changes in the format of input or output) without written approval from the Government. Should the program be furnished by the recipient to a third party, the recipient is responsible to that third party for any support and upkeep of the program. Information on the source of the program will be furnished to anyone requesting such information.

The accuracy of this program depends entirely on user-supplied data. It is the user's responsibility to understand how the input data affects the program output and to use the output data only as intended.

All documents and reports conveying information obtained as a result of the use of the program by the recipient will acknowledge the Corps of Engineers, Department of the Army, as the origin of the program. All such documentation will state the name and version of the program used by the recipient.

Accession For	
NTIS GRA&I <input checked="" type="checkbox"/>	
DTIC TAB <input type="checkbox"/>	
Unannounced <input type="checkbox"/>	
Justification _____	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special



A-1

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	September 1990	Final	
4. TITLE AND SUBTITLE  Scheduling for Cost-Plus Construction Projects, Volume II: PROMAN Source Files			5. FUNDING NUMBERS  MIPR CAB938, October 1988, "National Test Facility Consulting"
6. AUTHOR(S)  E. William East			8. PERFORMING ORGANIZATION REPORT NUMBER  USACERL TR P-90/31 Vol II
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  U.S. Army Construction Engineering Research Laboratory (USACERL) PO Box 4005 Champaign, IL 61824-4005			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  USACE, Omaha District 215 North 17th Street Omaha, NE 68102-4978			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES  Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words)  In cost-plus fixed fee contracts, the contractor is paid based on the actual costs incurred plus some graded incentive fee, rather than on a fixed bid. The larger the project, the more critical it becomes to ensure that the impact of all schedule changes be efficiently analyzed. In planning large projects, there is an acute need for supplemental software to process scheduling data to make it immediately useful to schedule reviewers. The PROMAN (PROject MANagement) microcomputer-based schedule analysis software was developed by the U.S. Army Construction Engineering Research Laboratory (USACERL) to provide several tools to help evaluate progress on Corps construction projects, assess the impact of schedule changes on the project, and provide other schedule management assistance.			
PROMAN was employed in the construction of the National Test Facility (NTF) in Colorado Springs, CO. Volume I of this report includes recommendations for the NTF Resident Office and future cost-plus projects to help evaluate progress schedules, and documentation for the installation and use of PROMAN. Volume II includes source code for the PROMAN software. The results of this study apply to all cost-plus construction projects and may also be applied to other large Corps construction projects.			
14. SUBJECT TERMS  PROMAN source files			15. NUMBER OF PAGES  202
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

## **FOREWORD**

This study was performed for the U.S. Army Corps of Engineers (USACE), Omaha District, under reimbursable project MIPR No. CAB938, October 1988, "National Test Facility Consulting." The project monitor was Mr. Aly Samahy, Resident Engineer at the National Test Facility Office, Colorado Springs, CO.

Special appreciation is expressed to the following members of the Engineering Staff at the National Test Facility: Mr. Larry Pierce, Mr. David Balding, and Ms. Veronica Rovera.

This work was performed by the Facility Systems Division (FS) of the U.S. Army Construction Engineering Research Laboratory (USACERL). Dr. Michael O'Connor is Chief of FS. The USACERL technical editor was William J. Wolfe, Information Management Office.

COL Everett R. Thomas is Commander and Director of USACERL, and Dr. L.R. Schaffer is Technical Director.

SCHEDULING FOR COST-PLUS CONSTRUCTION PROJECTS  
VOLUME II: PROMAN SOURCE FILES

```
*:*****  
*: Procedure file: PROCS.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/27/89      7:34  
*: Procs & Fncts: DISPERR  
*:          : CHECK  
*:          : ACTST  
*:          : CODEST  
*:          : PRECST  
*:          : PROGST  
*:          : FILENAME  
*:          : OPENFILS  
*:          : OPENNTFS  
*:          : CLOSEFIL  
*:          : TOTOP  
*:          : CERTIFY  
*: Documented 04/27/89 at 10:38           SNAP! version 3.12e  
*:*****  
* Procedure file for Activities to Crash Module
```

```

*|*****  

*!  

*!      Procedure: DISPERR  

*!  

*!      Called by: CHECK          (procedure in PROCS.PRG)  

*!                  : PAR_O.PRG  

*!                  : NTFST.PRG  

*!  

*!*****  

PROCEDURE disperr  

PARAMETERS code,filename  

*SET COLOR TO &errcol  

IF code = 1  

  @ ROW() +2,14 SAY filename+' has been used already -- WOULD YOU LIKE TO :'  

  *SET COLOR TO &menucol  

  @ ROW() +2,5 SAY ' 1) RETURN TO MAIN MENU?'  

  STORE 2 TO ch  

  @ ROW() +1,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?  SELECT 1 OR 2 : ' GET ch PICTURE '9'  

  READ  

  IF ch=1  

    option = 0  

  ENDIF  

ENDIF  

  

IF code = 2  

  @ ROW() +2,14 SAY filename+' has same modification/creation date as previous file.'  

  @ ROW() +1,5 SAY 'WOULD YOU LIKE TO : '  

  *SET COLOR TO &menucol  

  @ ROW() +2,5 SAY ' 1) RETURN TO MAIN MENU?'  

  STORE 2 TO ch  

  @ ROW() +1,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?  SELECT 1 OR 2 : ' GET ch PICTURE '9'  

  READ  

  IF ch=1  

    option = 0  

  ENDIF  

ENDIF  

  

IF code = 3  

  @ ROW() +2,14 SAY 'FILE '+filename+' DOES NOT EXIST.  WOULD YOU LIKE TO :'  

  *SET COLOR TO &menucol  

  @ ROW() +2,5 SAY ' 1) RETURN TO MAIN MENU?'  

  STORE 2 TO ch  

  @ ROW() +1,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?  SELECT 1 OR 2 : ' GET ch PICTURE '9'  

  READ  

  IF ch=1  

    option = 0  

  ENDIF  

ENDIF

```

```
IF code = 4
  @ ROW() +2,14 SAY 'FILE '+ filename +' IS NOT A DBF FILE. WOULD YOU LIKE TO :'
  *SET COLOR TO &menucol
  @ ROW() +2,5 SAY ' 1) RETURN TO MAIN MENU?'
  STORE 2 TO ch
  @ ROW() +1,5 SAY ' 2) INPUT AN ALTERNATE FILENAME? SELECT 1 OR 2 : ' GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
  ENDIF
ENDIF

IF code = 5
  @ ROW() +2,14 SAY 'FILE '+ filename +' IS NOT A SCEDULE FILE. WOULD YOU LIKE TO :'
  *SET COLOR TO &menucol
  @ ROW() +2,5 SAY ' 1) RETURN TO MAIN MENU?'
  STORE 2 TO ch
  @ ROW() +1,5 SAY ' 2) INPUT AN ALTERNATE FILENAME? SELECT 1 OR 2 : ' GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
  ENDIF
ENDIF
@ ROW() -11,1 CLEAR TO ROW(),79
RETURN
```

```

*!*****
*!
*!      Procedure: CHECK
*!
*!      Called by: PAR_O.PRG
*!
*!      Calls: DISPERR      (procedure in PROCS.PRG)
*!
*!      Uses: &FILENAME
*!
*!*****
PROCEDURE check
PARAMETERS filename,iserror

STORE LTRIM(RTRIM(filename)) TO filename
IF .NOT. FILE(filename)
  DO disperr WITH 3,filename
  STORE 1 TO iserror
ENDIF
IF iserror <> 1
  IF SUBSTR(filename,LEN(filename)-2,3) = 'DBF' .OR. SUBSTR(filename,LEN(filename)-3,3) = 'dbf'
    STORE 0 TO iserror
  ELSE
    DO disperr WITH 4,filename
    STORE 1 TO iserror
  ENDIF
ENDIF

IF iserror <> 1
  SELECT 9
  USE &filename
  IF FIELD(14, <> 'FFL'
    DO disperr WITH 5,filename
    STORE 1 TO iserror
  ENDIF
  USE
ENDIF

RETURN

```

```
*|*****  
*!  
*!      Procedure: ACTST  
*!  
*!      Called by: REPN.PRG  
*!  
*|*****  
PROCEDURE actst  
  
SELECT G  
APPEND BLANK  
REPL act WITH activ_id  
REPL TITLE WITH activ_dscr  
REPL od WITH activ_dur  
REPL dol_day WITH activ_cost/activ_dur  
STORE SUBSTR(start_net,3,3) TO mo  
REPL es WITH CTOD(&mo+'+'+SUBSTR(start_net,1,2)+ '/' +SUBSTR(start_net,6,2))  
STORE SUBSTR(finish_.at,3,3) TO mo  
REPL lf WITH CTOD(&mo+'+'+SUBSTR(finish_net,1,2)+ '/' +SUBSTR(finish_net,6,2))  
SELECT F  
  
RETURN
```

```
*!*****  
*!  
*!      Procedure: CODEST  
*!  
*!      Called by: REPN.PRG  
*!  
*!*****  
PROCEDURE codest  
  
SELECT G  
SEEK activ_id  
REPL resp WITH responsibl  
REPL claim WITH modnum  
REPL area WITH workarea  
REPL hammock WITH hamact  
REPL breakdown WITH breakdown  
REPL weather WITH weathers  
REPL TYPE WITH act_type  
REPL spec_sect WITH spec  
REPL buildsys WITH build  
REPL calcode WITH calcd  
SELECT F  
  
RETURN
```

```
*!*****  
*!  
*!      Procedure: PRECST  
*!  
*!      Called by: REPN.PRG  
*!  
*!*****  
PROCEDURE precst  
  
SELECT H  
APPEND BLANK  
REPL act WITH actidpr  
REPL suc WITH sucact  
REPL rel WITH suctype  
REPL lag WITH VAL(lagamt)  
SELECT F  
  
RETURN
```

```
*|*****  
*!  
*!      Procedure: PROGST  
*!  
*!      Called by: REPN.PRG  
*!  
*|*****  
PROCEDURE progst  
  
SELECT G  
SEEK actidprog  
REPL rd WITH VAL(remdur)  
STORE SUBSTR(actst,3,3) TO mo  
REPL as WITH CTOD(&mo+'/' +SUBSTR(actst,1,2)+'/' +SUBSTR(actst,6,2))  
STORE SUBSTR(actf,3,3) TO mo  
REPL af WITH CTOD(&mo+'/' +SUBSTR(actf,1,2)+'/' +SUBSTR(actf,6,2))  
SELECT F  
  
RETURN
```

```
*!*****  
*!  
*!      Procedure: FILENAME  
*!  
*!*****  
PROCEDURE filename  
PARAMETERS filename,wherex,wherey  
  
filename = TRIM(UPPER(filename))+"  
  
@ wherex,wherey  
@ wherex,wherey SAY "Enter Filename: " GET filename  
READ  
RETURN
```

```
*!*****  
*!  
*!      Procedure: OPENFILS  
*!  
*!      Called by: REPR.PRG  
*!          : REPS.PRG  
*!          : REPT.PRG  
*!          : REPU.PRG  
*!          : REPV.PRG  
*!          : REPW.PRG  
*!          : REPX.PRG  
*!  
*!      Calls: CLOSEFILE      (procedure in PROCS.PRG)  
*!  
*!      Uses: &NTFSCHED2  
*!          : &NTFSCHED1  
*!          : &NTFLOGIC2  
*!          : &NTFLOGIC1  
*!  
*!      Indexes: &NTFSCHED2  
*!          : &NTFSCHED1  
*!          : &NTFLOGIC2  
*!          : &NTFLOGIC1  
*!  
*!*****
```

```
PROCEDURE openfiles
```

```
SELECT 4  
USE  
DO closefil  
SELECT 1  
USE &ntfsched2 INDEX &ntfsched2  
SELECT 2  
USE &ntfsched1 INDEX &ntfsched1  
SELECT 5  
USE &ntflogic2 INDEX &ntflogic2  
SELECT 6  
USE ..ntflogic1 INDEX &ntflogic1  
RETURN
```

```

*|*****  

*!  

*!      Procedure: OPENNTFS  

*!  

*!      Called by: MAINX.PRG  

*!  

*!          Calls: CLOSEFIL      (procedure in PROCS.PRG)  

*!  

*!          Uses: &NTFSCHED2  

*!                  : &NTFSCHED1  

*!                  : &NTFLOGIC2  

*!                  : &NTFLOGIC1  

*!  

*!          Indexes: &NTFSCHED2  

*!                  : &NTFSCHED1  

*!                  : &NTFLOGIC2  

*!                  : &NTFLOGIC1  

*!  

*|*****  

PROCEDURE openntfs
CLEAR
@ 10,10 SAY "Generating indexes for reports R-X."  

  

ntfsched1 = UPPER(ntfsched1)
ntfsched2 = UPPER(ntfsched2)
ntflogic1 = UPPER(ntflogic1)
ntflogic2 = UPPER(ntflogic2)  

  

IF .NOT. AT(".DBF",ntfsched1) = 0
  ntfsched1 = STUFF(ntfsched1,AT(".",ntfsched1),4,"")
ENDIF  

  

IF .NOT. AT(".DBF",ntfsched2) = 0
  ntfsched2 = STUFF(ntfsched2,AT(".",ntfsched2),4,"")
ENDIF  

  

IF .NOT. AT(".DBF",ntflogic1) = 0
  ntflogic1 = STUFF(ntflogic1,AT(".",ntflogic1),4,"")
ENDIF  

  

IF .NOT. AT(".DBF",ntflogic2) = 0
  ntflogic2 = STUFF(ntflogic2,AT(".",ntflogic2),4,"")
ENDIF  

DO closefil
SELECT 1
USE &ntfsched2
INDEX ON act TO &ntfsched2
SELECT 2
USE &ntfsched1
INDEX ON act TO &ntfsched1
SELECT 5
USE &ntflogic2

```

```
INDEX ON act+suc TO &ntflogic2
SELECT 6
USE &ntflogic1
INDEX ON act+suc TO &ntflogic1
DO closefil
RETURN
```

```
* ! ****
*!
*!      Procedure: CLOSEFIL
*!
*!      Called by: OPENFILS      (procedure in PROCS.PRG)
*!                  : OPENNTFS      (procedure in PROCS.PRG)
*!                  : REPR.PRG
*!                  : REPS.PRG
*!                  : REPT.PRG
*!                  : REPV.PRG
*!                  : REPW.PRG
*!                  : REPX.PRG
*!
*! ****
PROCEDURE closefil

SELECT 1
USE
SELECT 2
USE
SELECT 5
USE
SELECT 6
USE

RETURN
```

```
*|*****  
*!  
*!      Procedure: TOTOP  
*!  
*!      Called by: TITLE1.PRG  
*!  
*|*****  
PROCEDURE totop  
  
* DO WHILE LINE < 65  
* ?  
*   LINE = LINE + 1  
* ENDDO  
* ?  
? eop // Form-Feed Character  
  
RETURN
```

```

* !*****
*!
*!      Procedure: CERTIFY
*!
*!      Called by: SETPARMS.PRG
*!
*!*****
PROCEDURE certify

PRIVATE temp

Q22,0 CLEAR TO 24,79

temp = 'Y'
Q 22,1 TO 22,79 DOUBLE
Q 23,1 SAY 'The values you have entered will be used for report '+TRIM(reportch)
Q 24,1 SAY 'Enter "Y" to accept, "N" to re-enter, or "Q" to quit: '
DO WHILE .T.
  Q 24,56 GET temp
  READ
  temp = TRIM(UPPER(temp))
  IF temp = 'Y'
    RETURN
  ELSE
    IF temp = 'N'
      option = 8
      RETURN
    ELSE
      IF temp = 'Q'
        option = 0
        RETURN
      ENDIF
    ENDIF
  ENDIF
ENDDO
RETURN
*: EOF: PROCS.PRG
^Z

```

```
*****  
*: Program: MAINVARS.PRG  
*:  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/27/89      0:22  
*:  
*: Called by: MAINMAIN.PRG  
*:  
*: Documented 04/27/89 at 10:38           SNAP! version 3.12e  
*****  
** MAINVARS.PRG - Declare and Initialize Public Variables for PROMAN  
*  
PUBLIC path1,path2,path3,path4,dataDate,menu  
*  
STORE '' TO path1,path2,path3,path4  && For Critical path strings  
dataDate = DATE()    && Primavera "Data Date", needed for Report M  
menu = ''  
*  
SET COLOR TO &menuCol  
*  
PUBLIC jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,Dec  
*  
jan='01'  && Strings for month translations in PSSR files  
feb='02'  
mar='03'  
apr='04'  
may='05'  
jun='06'  
jul='07'  
aug='08'  
sep='09'  
oct='10'  
nov='11'  
Dec='12'  
*  
PUBLIC code1,code8,code2,code3,code6,code7,code4,areaCodes  
*  
code1='STND'  && Activity Code or Activity Code ID used  
code8='RESP'  
*code5='WAD'  
code2='SORT'  
code3='SUBS'  
code6='AREA'  
code7='SUAR'  
code4='TYPE'  
* Search string for all "AREA" ID Codes in Schedule  
areaCodes='123456789'  
*  
* Strings for individual "AREA" ID Codes
```

```

*
PUBLIC areav1,areav2,areav3,areav4,areav5,areav6,areav7,areav8,areav9
*
areav1='Site Work'
areav2='Site Utilities'
areav3='Administration Building'
areav4='Research Building'
areav5='Central Plant'
areav6='ERROR - AREA 6 IS UNASSIGNED!'
areav7='Pass & ID/Entry Control Building'
areav8='Construction Support'
areav9='Construction Office'
*
PUBLIC stndcodes,stndv01,stndv02,stndv03,stndv04,stndv05,stndv06
PUBLIC stndv07,stndv08,stndv09,stndv10,stndv11,stndv12,stndv13,stndv14
PUBLIC stndv15,stndv16,stndvdr,stndvpm,stndvpr,stndvsc
*
* Search string for all "STND" ID Codes in Schedule
stndcodes='01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC'
* Strings for individual "STND" ID Codes
stndv01='General'
stndv02='Site Work'
stndv03='Concrete'
stndv04='Masonry'
stndv05='Metal'
stndv06='Wood and Plastics'
stndv07='Moisture-Thermal Control'
stndv08='Doors, Windows and Glass'
stndv09='Finishes'
stndv10='Specialties'
stndv11='Equipment'
stndv12='Furnishings'
stndv13='Special Construction'
stndv14='Conveying Systems'
stndv15='Mechanical'
stndv16='Electrical'
stndvdr='Drawings'
stndvpm='Prep Meetings'
stndvpr='Procurement'
stndvsc='Subcontract Award'
*
* Search string for all "SUB-AREA" ID Codes in Schedule
*
PUBLIC suarcodes,floorr0,floorr1,floorr2,floorr3,floorr4,sortcodes
*
suarcodes='0123456789'
floorr0='Basement'    && Strings for "SUB-AREA" Codes
floorr1='First Floor'
floorr2='Second Floor'
floorr3='Third Floor'
floorr4='Roof'
* Search string for all SORT Codes in Schedule (for Report M)

```

```

sortcodes='AA,AD,AF,AP,AW,CC,CD,CE,CF,CS,CW,DEMO,EC,ED,EE,EF,EG,EL,EP,EPL,ER,';
+'ES,ESL,ET,EU,EW,EX,FA,FF,FP,FW,ME,MH,MT,PB,PD,PF,PG,PI,PP,PR,PS,PW,SD,SM,';
+'SR,SS,UR'
*
PUBLIC srce,srss,srcc,srcf,sprr,srcw,spb,spd,spf,spg,spi,spes,spw,rad
PUBLIC srfa,sref,srew,sree,srmf,srfp
*
* Strings for individual "SORT" Codes (in rough order of construction progress)
srce='Earthwork '
ssrs='Structural Steel '
srcc='Concrete '
srcf='Concrete Floors '
sprr='Rough-In '
srcw='Dampproofing '
spb='Piping '
spd='Subdrains '
spf='Plumbing Fixtures '
spg='Gas Piping '
spi='Pipe Insulation '
spes='Sewer Piping '
spw='Water Piping '
rad='Drywall '
srfa='Finish Arch.
sref='Electrical Fixtures '
srew='Wire & Term.
sree='Electrical Equipment'
srmf='Mech. Equip.
srfp='Carpet & Lino.
srfp='Paint & Wallpaper '
*
PUBLIC respcodes,lvlscodes,typecodes,subscode1,subscode2
*
* Search string for all "RESP" Codes in Schedule
respcodes='ACEFMPSX'
* Search string for all "LVLS" Codes in Scehdule
lvlscodes='D,DC,DP,DZ,JO,M,OUT,S,Z'
* Search string for all "TYPE" ID Codes in Schedule
typecodes='00,10,20,30,40,60,70,99'
* Search string for all "SUBcontractS" Codes in Schedule
subscode1='ATR,CAFE,CARP3,CARP4,CEIL,CEIL4,CMU3,CMU4,COM3,COM4,COOL,CPCS,DOOR3,';
+'DOOR4,ELEV4,EMCS3,EMCS4,ENT,FD3,FD4,FE,FP3,FP4,GEN,HALO,HVAC3,HVAC4,LAND,PAVE,
subscode2='PI3,PI4,PID,PNT3,PNT4,RFC,RFLR3,RFLR4,RFT,ROOF3,ROOF4,SEC3,SEC4,SEW,STUD3,';
+'STUD4,T&B4,T&B4,TILE,WIND'
*
PUBLIC psr_done,sched_done,submi_done,processed,GET,output,print_ok
*
STORE .F. TO psr_done,sched_done,submi_done
processed = .F.
GET = ''   && For testing getpsr, getsched, etc with ON KEY
output = 'V'
print_ok = .F.
*

```

```
PUBLIC yn,PRINTER,maxlinesa,maxlinesf,maxlinesg,maxlinesh,maxlinesi  
PUBLIC maxlinesj,maxlinesk,maxlinesl,maxlinesn,maxlineso,eop,intro_on  
PUBLIC intro_off,header_on,header_mid,header_off,footer_on,footer_off  
*  
RETURN  
*: EOF: MAINVARS.PRG  
^Z
```

```

*:*****  

*: Program: SETFILES.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 20:20  

*: Called by: MAINMAIN.PRG  

*: Calls: FILEDISP.PRG  

*: Uses: FILELIST.DBF  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

PUBLIC filpssr_pr,filpssr_sc,filsched,filsucces,filsubmi  

SELECT 1  

USE filelist  

GOTO TOP  

filpssr_pr=pssr_proc  

filpssr_sc=pssr_subs  

filsched=schedule  

filsucces=successor  

filsubmi=submittal  

USE  

CLEAR  

@2,2 SAY " "  

TEXT  

The files listed as defaults below indicate the last stored values for  

Submittal, Schedule, and Successor databases. These are typically either  

the last set selected by a PROMAN user or the last files processed by the  

system. If a file is marked as non-existent, it is mandatory that an  

existing file be entered before attempting to run reports. The PSSR  

files required for some reports are not displayed below. If new PSSR  

records for either procurements or subcontracts are moved to the PROMAN  

directory, use File Processing Options to create new PSSR databases.  

Indexes for each of the databases will be created if the files are  

required for selected reports or operations.  

ENDTEXT  

DO filedisp  

@14,2 SAY " "  

WAIT  

RETURN  

  

*: EOF: SETFILES.PRG  

^Z

```

```

*:*****  

*: Program: MENUONE.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/25/89      13:05  

*: Called by: MAINMAIN.PRG  

*: Calls: FILEDISP.PRG  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

option = 0  

temp = 0  

temp1 = "0"  

CLEAR  

@ 1,30 SAY "WELCOME TO PROMAN"  

@ 2,30 SAY "=====  

@ 5,18 TO 14,57 DOUBLE  

@ 6,22 SAY '1 - Run Batch Reports'  

@ 7,22 SAY '2 - ASCII Standard File Import'  

@ 8,22 SAY '3 - List Database Files'  

@ 9,22 SAY '4 - Process New Project Files'  

@ 10,22 SAY '5 - Change Active Project Files'  

@ 12,22 SAY '9 - Exit Program'  

DO filedisp  

@ 17,22 SAY 'Please enter your selection.'  

  

DO WHILE option = 0  

  DO WHILE temp = 0  

    temp = INKEY()  

  ENDDO  

  temp1 = UPPER(CHR(temp))  

  DO CASE  

    CASE temp1 = "1"  

      option = 1  

    CASE temp1 = "2"  

      option = 3  

    CASE temp1 = "3"  

      option = 4  

    CASE temp1 = "4"  

      option = 5  

    CASE temp1 = "5"  

      option = 6  

    CASE temp1 = "9"  

      option = 9  

  ENDCASE

```

```
temp = 0
ENDDO
RETURN
*: EOF: MENUONE.PRG
^Z
```

```

*:*****  

*: Program: MENUTWO.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      22:53  

*: Called by: MAINMAIN.PRG  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

SET TALK OFF
reportst = ''
exitflag = .F.
CLEAR
@1,5 SAY "PROMAN REPORTS"
@2,5 SAY "===="
@0,31 TO 22,79 DOUBLE
@1,35 SAY 'A - PSSR/Schedule Matching'
@2,35 SAY "B - PSSR/Schedule 'PR' Code Matching"
@3,35 SAY 'C - PSSR/Schedule; Date Comparison 1'
@4,35 SAY 'D - PSSR/Schedule; Date Comparison 2'
@5,35 SAY 'E - PSSR/Schedule/Submittal Comparison 1'
@6,35 SAY 'F - PSSR/Schedule/Submittal Comparison 2'
@7,35 SAY 'G - Resource Loading - Manpower Variance'
@8,35 SAY 'H - Resource Loading - Funding Variance'
@9,35 SAY 'I - Resource Loading - Time Variance'
@10,35 SAY 'J - Logical Analysis'
@11,35 SAY 'K - Activities to Crash'
@12,35 SAY 'L - Delays in Design'
@13,35 SAY 'M - Contractor Grading'
@14,35 SAY 'O - Activity Slippage'
@15,35 SAY 'R - Deleted Activities'
@16,35 SAY 'S - Find Added Activities'
@17,35 SAY 'T - Find PCT AS/AF Inconsistency'
@18,35 SAY 'U - Find Changes in Description'
@19,35 SAY 'V - Find Changes in Orig. Duration'
@20,35 SAY 'W - Find Changes in Progress'
@21,35 SAY 'X - Find Logic Changes'
@23,35 SAY '*' - indicates selected reports'
@17,5 SAY ' OTHER ACTIONS '
@18,5 SAY "===="
@19,5 SAY '1 - RUN REPORTS'
@20,5 SAY '2 - PREVIOUS MENU'
@21,5 SAY '9 - QUIT'
@4,5 SAY 'REPORT SELECTION'
@6,5 SAY 'Press a letter to'
@7,5 SAY 'choose reports to'
@8,5 SAY 'run.'
@11,5 SAY 'Your Selection'

```

```

@12,5 SAY ">"
DO WHILE .NOT. exitflag
  rsval = 0
  DO WHILE rsval = 0
    rsval = INKEY()
  ENDDO
  rs = UPPER(CHR(rsval))
  scrpos = 0
  DO CASE
    CASE rs >= "A" .AND. rs < "N"
      scrpos = ASC(rs) - 64
    CASE rs = "O"
      scrpos = 14
    CASE rs > "P" .AND. rs < "Y"
      scrpos = ASC(rs) - 67
    CASE rs = "1"
      CLEAR
      RETURN
    CASE rs = "2"
      CLEAR
      reportst = ""
      RETURN
    CASE rs = "9"
      exitflag = .T.
    OTHERWISE
    ENDCASE
    IF scrpos > 0
      temp = AT(rs,reportst)
      IF temp = 0
        reportst = reportst + rs
        @ scrpos,33 SAY "*"
      ELSE
        reportst = STUFF(reportst,temp,1,"")
        @ scrpos,33 SAY " "
      ENDIF
      scrpos = 0
    ENDIF
    @ 12,5 SAY '
    @ 12,5 SAY '>'
  ENDDO
*: EOF: MENUTWO.PRG
^Z

```

```
*:*****  
*: Program: MAINX.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/27/89 9:09  
*: Called by: MAINMAIN.PRG  
*: Calls: PRINTSET.PRG  
*: : SETPARMS.PRG  
*: : KEYPRESS.PRG  
*: : OPENNTFS (procedure in PROCS.PRG)  
*: : INTRO.PRG  
*: : REPA.PRG  
*: : REPB.PRG  
*: : REPC.PRG  
*: : REPD.PRG  
*: : REPE.PRG  
*: : REPF.PRG  
*: : REPG.PRG  
*: : REPH.PRG  
*: : REPI.PRG  
*: : REPJ.PRG  
*: : REPK.PRG  
*: : REPL.PRG  
*: : REPM.PRG  
*: : REPO.PRG  
*: : REPR.PRG  
*: : REPS.PRG  
*: : REPT.PRG  
*: : REPU.PRG  
*: : REPV.PRG  
*: : REPW.PRG  
*: : REPX.PRG  
*: Uses: PSSR_PR.DBF  
*: : &FILSCHED  
*: : &FILSUBMI  
*: : ZZZTEMP.DBF  
*: : &FILSUCCES  
*: : &FILSCH2L  
*: : &FILSCH1L  
*: : GRADING1.DBF  
*: : GRADING2.DBF  
*: Indexes: PSSR_ACT.NDX  
*: : SCHD_STN.NDX  
*: : SCHD_ACT.NDX  
*: : SUBM_NAS.NDX
```

```

*:          : SCH2_ACT.NDX
*:          : SCH1_ACT.NDX
*:
*:      Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

** MAIN.PRG - Start Pro-Man
SET COLOR TO &menucol

IF 'A'$reportst.OR.'B'$reportst.OR.'C'$reportst.OR.'D'$reportst.OR.'E'$reportst;
    .OR.'F'$reportst.OR.'M'$reportst.OR.'N'$reportst
IF .NOT. FILE("pssr_pr.dbf")
    CLEAR
    @ 10,10 SAY "REPORTS A,B,C,D,E,F, AND M REQUIRE PSSR DATA."
    @ 11,10 SAY "YOU MUST USE THE FILE PROCESSING OPTION TO CREATE"
    @ 12,10 SAY "VALID PSSR_PR.DBF FILE OR YOU MUST COPY THE FILE"
    @ 13,10 SAY "TO THE PROMAN SUBDIRECTORY BEFORE ATTEMPTING TO"
    @ 14,10 SAY "RUN THESE REPORTS. YOU WILL BE RETURNED TO THE"
    @ 15,10 SAY "MAIN MENU."
    option = 0
    reportst = ""
    RETURN
ENDIF
ENDIF
*
DO printset
* printer option selection

STORE .y. TO pssr_done,sched_done,submi_done           && <*****
processed=.y.                                         && <*****

* Prompt user for Report-specific responses:

STORE '1' TO rgcode,rhcode,ricode
STORE 10 TO refloat,ufloat,upercent
STORE 20.0 TO rfpercent
STORE 20 TO rgmn,durat,reldur
STORE 1.00 TO rgsd
STORE 2000 TO rhdl
STORE 1 TO rhsd,risd
STORE 40 TO ridy
STORE '2' TO rjcode
STORE 5 TO mpl
STORE SPACE(20) TO filschil,filsched2,filsched3,filsched4
STORE SPACE(20) TO filsched5,ntfsched2,ntflogic2
STORE filsched+SPACE(12) TO filsch21,filsched1,ntfsched1
STORE f'lsuccess+SPACE(12) TO ntflogici
STORE 'DR' TO stnd
STORE 'S' TO mch
STORE CTOD(' / / ') TO datadate
STORE 0 TO numfiles
*
```

```

* get the parameters for reports
*
DO setparms
IF option = 0
    RETURN
ENDIF

* Tell user about his output filenames
CLEAR
nn=1
STORE '' TO reportch
STORE 0 TO pdo
DO WHILE nn<=LEN(reportst)
    IF (SUBSTR(reportst,nn,1)<>'N') .AND. (SUBSTR(reportst,nn,1)<>'P') .AND.
(SUBSTR(reportst,nn,1)<>'Q')
        STORE reportch + SUBSTR(reportst,nn,1) TO reportch
    ENDIF
    IF (SUBSTR(reportst,nn,1)='P')
        STORE 1 TO pdo
    ENDIF
    nn=nn+1
ENDDO
STORE SUBSTR(reportch,2,LEN(reportch)-1) TO reportst
CLEAR
@3,10 SAY "Changing old SASREP.TXT names to SASREP.OLD."
RUN del sasrep*.old
RUN ren sasrep*.txt sasrep*.old
IF (output='B'.OR.output='D'.OR.output='A') .AND. (LEN(reportst)>0)
    fname='SASREP'
    @3,10 SAY 'Your output filenames will be :'
    nn=1
    reportch=' '
    DO WHILE nn<=LEN(reportst)
        reportch=SUBSTR(reportst,nn,1)
        ? SPACE(15)+'&fname&reportch .TXT'
        nn=nn+1
    ENDDO
    WAIT
ENDIF
* Set keyboard "off"
SET ESCAPE OFF
ON KEY DO keypress
* Do Initial file processing
SET STATUS ON
CLEAR
@10,10 SAY "Setting up files for All Reports."
IF 'A'$reportst.OR.'B'$reportst.OR.'C'$reportst.OR.'D'$reportst.OR.'E'$reportst;
    .OR.'F'$reportst.OR.'M'$reportst.OR.'N'$reportst
SELECT B    && pssr_PR.dbf - Now merged with pssr_SC.dbf
USE
USE pssr_pr
INDEX ON activ_id TO pssr_act

```

```

ENDIF
SELECT D    && filsched
USE &filsched
IF 'G'$reportst .OR. 'H'$reportst           .OR. 'I'$reportst
    INDEX ON SUBSTR(act,5,2) TO schd_stn
ENDIF
INDEX ON act TO schd_act

IF 'E'$reportst.OR.'F'$reportst.OR.'M'$reportst.OR.'N'$reportst
    SELECT E
    USE &filsubmi
    INDEX ON SUBSTR(nas,1,9) TO subm_nas
ENDIF
*
IF 'R'$reportst.OR.'S'$reportst.OR.'T'$reportst.OR.'U'$reportst.OR.'V'$reportst;
    .OR.'W'$reportst.OR.'X'$reportst
    CLOSE DATABASES
    DO openntfs
ENDIF

*
nn=0      && Generate all Reports
reportch='A'
DO WHILE nn <= 26
    IF reportch $ reportst
        IF output='B'.OR.output='D'.OR.output='A'
            SET ALTERNATE TO &fname&reportch
            SET ALTERNATE ON
        ENDIF
        IF output='P'.OR.output='B'
            SET PRINT ON
        ENDIF
        CLOSE DATABASES
        SELECT D
        USE &filsched
        SELECT B
        USE pssr_pr
        DO intro      && Do Intro to each Report
        SELECT D
        USE
        SELECT B
        USE
        DO CASE          && Replaced Macro-substitution for compiler purposes
        CASE reportch='A'
            SELECT B
            USE pssr_pr INDEX pssr_act
            SELECT D
            USE &filsched INDEX schd_act
            DO repa
            SELECT B
            USE
            SELECT D

```

```
USE
CASE reportch='B'
  SELECT B
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  DO repb
  SELECT B
  USE
  SELECT D
  USE
CASE reportch='C'
  SELECT B
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  DO repc
  SELECT B
  USE
  SELECT D
  USE
CASE reportch='D'
  SELECT B
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  DO repd
  SELECT B
  USE
  SELECT D
  USE
CASE reportch='E'
  SELECT B
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  SELECT E
  USE &filsubmi INDEX subm_nas
  DO repe
  SELECT B
  USE
  SELECT D
  USE
  SELECT E
  USE
CASE reportch='F'
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  SELECT E
  USE &filsubmi INDEX subm_nas
  DO repf
```

```

SELECT B
USE
SELECT D
USE
SELECT E
USE
CASE reportch='G'
  SELECT D
  USE &filsched INDEX schd_stn
  DO repg
  SELECT D
  USE
CASE reportch='H'
  SELECT D
  USE &filsched INDEX schd_stn
  DO repb
  SELECT D
  USE
CASE reportch='I'
  SELECT D
  USE &filsched INDEX schd_stn
  DO repi
  SELECT D
  USE
CASE reportch='J'
  IF FILE("zzztemp.dbf")
    ERASE zzztemp.dbf
  ENDIF
  COPY FILE &filsched TO zzztemp.dbf
  SELECT D
  USE zzztemp
  DO repj
  SELECT D
  USE
  ERASE zzztemp.dbf
CASE reportch='K'
  SELECT B
  USE &filsucces
  SELECT D
  USE &filsched
  DO repk
  SELECT B
  USE
  SELECT D
  USE
CASE reportch='L'
  SELECT G
  USE &filsch21
  INDEX ON act TO sch2_act
  SELECT F
  USE &filsch11
  INDEX ON SUBSTR(act,5,2) TO schl_act

```

```

DO REPL
SELECT G
USE
SELECT F
USE
CASE reportch='M'
  SELECT F
  USE grading1
  SELECT G
  USE grading2
  SELECT B
  USE pssr_pr INDEX pssr_act
  SELECT D
  USE &filsched INDEX schd_act
  SELECT E
  USE &filsubmi INDEX subm_nas
  DO repm
  SELECT G
  USE
  SELECT F
  USE
  SELECT B
  USE
  SELECT D
  USE
  SELECT E
  USE
CASE reportch='O'
  IF numfiles > 1
    DO REPO
  ENDIF
CASE reportch='R'
  DO repr
CASE reportch='S'
  DO reps
CASE reportch='T'
  DO rept
CASE reportch='U'
  DO repu
CASE reportch='V'
  DO repv
CASE reportch='W'
  DO repw
CASE reportch='X'
  DO repx
ENDCASE
IF output='B'.OR.output='D'.OR.output='A'
  ?ecp
  CLOSE ALTERNATE
ENDIF
IF output='P'.OR.output='B'
  ?eop

```

```
SET PRINT OFF
ENDIF
ENDIF
nn=nn+1
reportch = CHR(ASC('A')+nn)
ENDDO
*: EOF: MAINX.PRG
```

```
*:*****  
*: Program: REPN1.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/17/90 16:25  
*: Called by: MAINMAIN.PRG  
*: Calls: QUIT.PRG  
*: : REPN.PRG  
*: Documented 04/27/89 at 10:38 SNAP! version 3.12e  
*:*****
```

```
SET CONFIRM ON  
CLEAR  
Q 1,20 SAY 'REPORT N: ASCII INTERFACE'  
Q 3,4 SAY ''  
TEXT
```

This routine converts project schedule data from an ASCII file format to the formats used in the Scheduling Analysis System (SAS). The ASCII file must have been previously been created in the format specified in the "Data Exchange Standard Goal" publication (This standard is designed as a standardized logical file format, into which any Project Management Software which has the ability to export data in ASCII format, can save data for each project into a single file.)

Refer to the SAS User Manual for instructions regarding how to create this ASCII file, or for more detailed information refer to the paper cited above.

If this ASCII file has not yet been created, you will need to exit SAS and create it.

```
ENDTEXT
```

```
exitnow=''  
Q20,15 SAY 'RETURN TO MAIN MENU NOW <Y/N> ? ' GET exitnow PICTURE '!'  
READ  
IF exitnow='Y'  
  RETURN  
ENDIF  
WAIT  
  
CLEAR  
Q 3, 5 SAY ' Assuming that this ASCII file has been created, you will be prompted'  
Q 4, 5 SAY 'to enter the name of that file, and it will be converted to the file'  
Q 5, 5 SAY 'formats required by SAS. This report will also ask you for the names '  
Q 6, 5 SAY 'of the schedule and successor files to create. If these already exist '  
Q 7, 5 SAY 'you will be given the option to exit and remove the files or choose a new'  
Q 8, 5 SAY 'file name.'
```

```

STORE SPACE(12) TO asciifile
DO WHILE .T.
  @9,5 SAY 'INPUT THE NAME OF THE ASCII FILE CONTAINING PROJECT DATA : ' GET asciifile PICTURE
'!!!!!!!!!!!!'
  READ
  IF asciifile = ''
    EXIT
  ENDIF
  IF .NOT. FILE(asciifile)
    @10,10 SAY 'THAT FILENAME DOES NOT EXIST IN THE CURRENT DIRECTORY. IF YOU HAVE'
    @11,10 SAY 'MISTYPED THE NAME, PLEASE TYPE IT AGAIN. HOWEVER, IF YOU HAVE TYPED'
    @12,10 SAY 'THE CORRECT NAME, YOU MUST EXIT SAS AND CREATE THIS FILE.'
    exitnow= ''
    @14,15 SAY 'EXIT SAS NOW <Y/N> ? ' GET exitnow PICTURE '!'
    READ
    IF exitnow='Y'
      DO QUIT
    ENDIF
  ELSE
    EXIT
  ENDIF
ENDDO
IF asciifile <> ''
  DO WHILE .T.
    STORE SPACE(8) TO schdfil
    @ 15,5 SAY 'INPUT THE NAME OF THE SCHEDULE FILE TO CREATE : ' GET schdfil PICTURE '!!!!!!!!'
    @ 16,5 SAY 'DO NOT INCLUDE THE .DBF ENDING ON THE NAME.'
    @ 17,5 SAY 'ENTER A SPACE IF YOU DO NOT WISH TO CREATE A SCHEDULE FILE'
    READ
    IF FILE(schdfil)
      @ 18,10 SAY 'THAT FILENAME ALREADY EXISTS. PLEASE USE A DIFFERENT NAME OR DELETE'
      @ 19,10 SAY 'THE CURRENT FILE WITH THAT NAME.'
      exitnow = ''
      @ 20,10 SAY 'EXIT SAS NOW <Y/N> ? ' GET exitnow PICTURE '!'
      READ
      IF exitnow = 'Y'
        DO QUIT
      ENDIF
    ELSE
      EXIT
    ENDIF
    @ 15,5 CLEAR TO 19,79
  ENDDO

@ 15,5 CLEAR TO 16,79

DO WHILE .T.
  STORE SPACE(8) TO sucfil
  @ 15,5 SAY 'INPUT THE NAME OF THE SUCCESSOR FILE TO CREATE : ' GET sucfil PICTURE '!!!!!!!!'
  @ 16,5 SAY 'DO NOT INCLUDE THE .DBF ENDING ON THE NAME.'
  READ

```

```

IF FILE(sucfil)
  @ 17,10 SAY 'THAT FILENAME ALREADY EXISTS. PLEASE USE A DIFFERENT NAME OR DELETE'
  @ 18,10 SAY 'THE CURRENT FILE WITH THAT NAME.'
  exitnow = ''
  @ 19,10 SAY 'EXIT SAS NOW <Y/N> ? ' GET exitnow PICTURE '!'
  READ
  IF exitnow = 'Y'
    DO QUIT
  ENDIF
ELSE
  EXIT
ENDIF
ENDDO
@ 15,5 CLEAR TO 19,79
ENDIF
CLEAR
STORE SPACE(12) TO submitfile
DO WHILE .T.
  @9,5 SAY 'INPUT THE NAME OF THE ASCII FILE CONTAINING SUBMITTAL DATA : ' GET submitfile PICTURE
'!!!!!!!!!!!!'
  READ
  IF submitfile = ''
    EXIT
  ENDIF
  IF .NOT. FILE(submitfile)
    @10,10 SAY 'THAT FILENAME DOES NOT EXIST IN THE CURRENT DIRECTORY. IF YOU HAVE'
    @11,10 SAY 'MISTYPED THE NAME, PLEASE TYPE IT AGAIN. HOWEVER, IF YOU HAVE TYPED'
    @12,10 SAY 'THE CORRECT NAME, YOU MUST EXIT SAS AND CREATE THIS FILE.'
    exitnow=' '
    @14,15 SAY 'EXIT SAS NOW <Y/N> ? ' GET exitnow PICTURE '!'
    READ
    IF exitnow='Y'
      DO QUIT
    ENDIF
  ELSE
    EXIT
  ENDIF
ENDDO
IF submitfile <> ''
DO WHILE .T.
  STORE SPACE(8) TO subfil
  @ 15,5 SAY 'INPUT THE NAME OF THE SUBMITTAL FILE TO CREATE : ' GET subfil PICTURE '!!!!!!!!'
  @ 16,5 SAY 'DO NOT INCLUDE THE .DBF ENDING ON THE NAME.'
  READ
  IF FILE(subfil)
    @ 18,10 SAY 'THAT FILENAME ALREADY EXISTS. PLEASE USE A DIFFERENT NAME OR DELETE'
    @ 19,10 SAY 'THE CURRENT FILE WITH THAT NAME.'
    exitnow = ''
    @ 20,10 SAY 'EXIT SAS NOW <Y/N> ? ' GET exitnow PICTURE '!'
    READ
    IF exitnow = 'Y'
      DO QUIT
  ENDIF

```

```
ENDIF
ELSE
  EXIT
ENDIF
  @ 15,5 CLEAR TO 20,79
ENDDO
ENDIF
IF asciiifile <> ''
  CLEAR
  @ 3, 5 SAY 'The ASCII Interface Module will use '+asciiifile+' for the input file.'
  @ 4, 5 SAY 'schdfilt' will be used for the schedule database, and '+sucfilt' will'
  @ 5, 5 SAY 'be used as the successor database.'
  STORE '' TO exitnow
ENDIF
SET CONFIRM OFF
DO repn
RETURN
*: EOF: RPPN1.PRG
^Z
```

```

*:*****  

*: Program: MENUTHRE.PRG  

*.  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      0:55  

*: Called by: MAINMAIN.PRG  

*: Calls: FILEDISP.PRG  

*:       : GETSCHED.PRG  

*:       : SCHEDULE.PRG  

*:       : GETSUBMI.PRG  

*:       : SUBMIT.PRG  

*:       : GETPSSR.PRG  

*:       : PSSR.PRG  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

option = 0  

temp = 0  

temp1 = "0"  

CLEAR  

@ 2,20 SAY "FILE PROCESSING OPERATIONS"  

@ 4,4 SAY ""  

TEXT  

PROMAN can be used with files that are provided in ASCII, according to  

published standard file formats (see users manual) or files can be  

processed from the Primavera project manager program. If files are  

imported to the system from Primavera, the files must be processed prior  

to use with PROMAN reports. The processing steps need be repeated only  

once for each type of file. A menu is provided to enable processing of  

each of the file types or to process all types at once. Processing  

should be performed whenever new data files are added to the PROMAN area  

for analysis. Anytime a new file is processed, it is made the default  

file name for the most current files in the report modules. As a result,  

users need only press return to select a file for most current information  

in most reports.  

ENDTEXT  

?  

WAIT  

DO WHILE option = 0  

CLEAR  

@ 1,27 SAY "PROMAN FILE PROCESSOR MENU"  

@ 2,27 SAY '===== "  

@ 5,19 TO 14,63 DOUBLE  

@ 6,21 SAY '1 - Process Schedule and Successor Files'  

@ 7,21 SAY '2 - Process a Submittal File'  

@ 8,21 SAY '3 - Process Subcontract/Procurement files'  

@ 9,21 SAY '4 - Process Items 1, 2, and 3'

```

```

@ 10,21 SAY '5 - List Database Files'
@ 11,21 SAY '6 - List Files with "PSSR" Prefixes'
@ 13,21 SAY '9 - Exit to Main Menu'
@ 17,21 SAY 'Please enter your selection.'

DO filedisp

DO WHILE temp = 0
    temp = INKEY()
ENDDO
temp1 = UPPER(CHR(temp))
DO CASE
CASE temp1 = "1"
    option = 1
    DO getsched
    IF option > 0
        SET STATUS ON
        DO schedule
        SET STATUS OFF
    ENDIF
    option = 0
CASE temp1 = "2"
    option = 2
    DO getsubmi
    IF option > 0
        SET STATUS ON
        DO submit
        SET STATUS OFF
    ENDIF
    option = 0
CASE temp1 = "3"
    option = 3
    sc_exist = .F.
    DO getpsr
    IF option > 0
        SET STATUS ON
        DO psr
        SET STATUS OFF
    ENDIF
    option = 0
CASE temp1 = "4"
    option = 4
    DO getsched
    IF option > 0
        DO getsubmi
    ENDIF
    IF option > 0
        sc_exist = .F.
        DO getpssr
    ENDIF
    IF option > 0
        SET STATUS ON

```

```
DO schedule
DO submit
DO pssr
SET STATUS OFF
ENDIF
option = 0
CASE templ = "5"
CLEAR
DIR *.dbf
WAIT
CASE templ = "6"
CLEAR
DIR pssr.*.*
WAIT
CASE templ = "9"
option = 9
ENDCASE
temp = 0
ENDDO
RETURN
```

\*: EOF: MENUTHRE.PRG  
^Z

```

*:*****
*: Program: CHGNAME.PRG
*:
*: System: PROJECT MANAGEMENT TOOLS
*: Author: ASC/RCE and USA-CERL
*: Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      0:56
*:
*: Called by: MAINMAIN.PRG
*:
*: Calls: FILEDISP.PRG
*:
*: Uses: FILELIST.DBF
*:
*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****
CLEAR
temp = 0
DO WHILE option > 0
  filsched = filsched + "
  filsucces = filsucces + "
  filsubmi = filsubmi + "
  @ 8,5 SAY "Change the active file names"
  @ 10,10 SAY "Active Schedule File = " GET filsched
  @ 11,10 SAY "Active Successor File = " GET filsucces
  @ 12,10 SAY "Active Submittal File = " GET filsubmi
READ
IF .NOT. FILE(filsched)
  temp = temp + 1
ENDIF
IF .NOT. FILE(filsucces)
  temp = temp + 1
ENDIF
IF .NOT. FILE(filsubmi)
  temp = temp + 1
ENDIF
IF temp = 0
  SELECT 1
  USE filelist
  GOTO TOP
  REPLACE schedule WITH UPPER(TRIM(filsched))
  REPLACE successor WITH UPPER(TRIM(filsucces))
  REPLACE submittal WITH UPPER(TRIM(filsubmi))
  option = 0
  USE
  RETURN
ENDIF
@ 20,1 CLEAR TO 24,79
DO filedisp
temp = 0
ENDDO

*: EOF: CHGNAME.PRG
^Z

```

```

*:*****
*:
*:      Program: FILEDISP.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/20/89      22:43
*:
*:      Called by: SETFILES.PRG
*:                  : MENUONE.PRG
*:                  : MENUTHRE.PRG
*:                  : CHGNAME.PRG
*:
*:      Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****
temp1 = " <- Does Not Exist "
temp2 = temp1
temp3 = temp1
IF FILE(filsched)
  temp1 = " "
ENDIF
IF FILE(filsucces)
  temp2 = " "
ENDIF
IF FILE(filsubmi)
  temp3 = " "
ENDIF
021,20 SAY "Active Schedule File = " + filsched
022,20 SAY "Active Successor File = " + filsucces
023,20 SAY "Active Submittal File = " + filsubmi
021,57 SAY temp1
022,57 SAY temp2
023,57 SAY temp3
020,18 TO 24,57
RETURN
*: EOF: FILEDISP.PRG
^Z

```

```

*:*****  

*: Program: PRINTSET.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/20/89      16:33  

*: Called by: MAINX.PRG  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e  

*:*****  

  

SET COLOR TO &menucol  

  

output='P'  

@1,20 SAY 'CHOOSE OUTPUT METHOD:'  

@4,10 SAY 'P Print Report'  

@5,10 SAY 'B Print AND Make Disk File of Report WITH PRINTER CODES'  

@6,10 SAY 'D Make Disk File of Report WITH PRINTER CODES'  

@7,10 SAY 'A Make Disk File of Report WITHOUT PRINTER CODES'  

DO WHILE .T.  

@9,15 SAY 'SELECT ONE OF THESE OPTIONS <P,B,D or A>: ' GET output PICTURE ''!  

READ  

@11,0  

yn=.n.  

DO CASE  

CASE output$'PBDA'  

@11,20 SAY 'VALID OUTPUT CHOICE.'  

EXIT  

OTHERWISE  

SET COLOR TO &errcol  

@11,20 SAY 'NOT A VALID OPTION - SELECT AGAIN!'  

SET COLOR TO &menucol  

LOOP  

ENDCASE  

ENDDO  

IF output='P'.OR.output='B'.OR.output='D'  

@13,10 SAY 'Even if you chose to direct your Reports only to a disk file,'  

@14,10 SAY 'information about the printer on which this disk file will eventually'  

@15,10 SAY 'be printed is required to properly format the output, IF you will be'  

@16,10 SAY 'sending the file directly to a printer.'  

@18,15 SAY 'Please identify your printer type:'  

@19,20 SAY 'W Epson Wide-Carriage or Compatible'  

@20,20 SAY 'L Hewlett-Packard LaserJet or LaserJet 2'  

PRINTER='L'  

DO WHILE .T.  

@22,15 SAY 'Enter Your Choice: ' GET PRINTER PICTURE ''!  

READ  

@23,0  

IF .NOT.PRINTER$'WL'

```

```

SET COLOR TO &errcol
@24,20 SAY 'INCORRECT CHOICE - PLEASE RE-ENTER'
SET COLOR TO &menucol
ELSE
  EXIT
ENDIF
ENDDO
DO CASE
CASE PRINTER='L'
  maxlinesa=53  && Maximum no. data lines allowed per page, for each Report
  maxlinesf=53
  maxlinesg=53
  maxlinesh=53
  maxlinesi=53
  maxlinesj=53
  maxlinesk=53
  maxlinesl=53
  maxlinesn=53
  maxlineeo=53
CASE PRINTER='W'
  maxlinesa=53
  maxlinesf=53
  maxlinesg=53
  maxlinesh=53
  maxlinesi=53
  maxlinesj=53
  maxlinesk=53
  maxlinesl=53
  maxlinesn=53
  maxlineeo=53
ENDCASE
eop=CHR(12)
DO CASE  && Printer setup strings for Reports
CASE PRINTER='L'
  intro_on="CHR(27)+'&110'+CHR(27)+'&17.0C'+CHR(27)+' (s4.16H'+CHR(27)+'&k18.7H'+CHR(27)+' (s3B' "
  *           Landscape      ?      Character size/BOLD(?)   Spacing of letters Bold ON
  intro_off=  "CHR(27)+'&15.57C'+CHR(27)+'&k9.35H'+CHR(27)+'&l6D'+CHR(27)+' (s0B' "
  header_on="CHR(27)+'&17.0C'+CHR(27)+' (s4.16H'+CHR(27)+'&k18.7H'+CHR(27)+' (s3B'+CHR(27)+'&l8D' "
  *           ?           font size?      spacing of letters Bold ON     8 lines/inch
  header_mid="CHR(27)+' (s3B'+CHR(27)+'&15.57C'+CHR(27)+' (s16.67H'+CHR(27)+'&k9.35H' "
  header_off="CHR(27)+' (s16.67H'+CHR(27)+'&l8D' "
  footer_on="CHR(27)+'&15.57C'+CHR(27)+' (s16.67H'+CHR(27)+'&k9.35H' "
  footer_off="CHR(27)+' (s16.67H' "
CASE PRINTER='W'
  intro_on="CHR(27)+CHR(64)+CHR(27)+'W1'+CHR(27)+'E'" && Reset, ON Expanded, Emphasized
  intro_off="CHR(27)+'W0'+CHR(27)+'F'+CHR(27)+'M'" && OFF Expanded, Emphasized, ON Elite
  header_on="CHR(27)+CHR(64)+CHR(27)+CHR(18)+CHR(27)+'P'+CHR(27)+'E'" && OFF Compressed, ON Pica,
  Emphasized
  header_mid="CHR(27)+'F'" && OFF Emphasized
  header_off="CHR(27)+CHR(15)" && ON Compressed
  footer_on="CHR(27)+CHR(18)+CHR(27)+'M'" && OFF Compressed, ON Elite
  footer_off="CHR(27)+'P'" && ON Pica (OFF Elite)

```

```
    ENDCASE
ENDIF
IF output='A'
maxlinesa=53
maxlinesf=53
maxlinesg=53
maxlinesh=53
maxlinesi=53
maxlinesj=53
maxlinesk=53
maxlinesl=53
maxlinesn=53
maxlineso=53
intro_on=''
intro_off=''
header_on=''
header_mid=''
header_off=''
footer_on=''
footer_off=''
*
eop=''

ENDIF
*
RETURN
*: EOF: PRINTSET.PRG
^Z
```

```

*:*****  

*: Program: SETPARMS.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      16:31  

*: Called by: MAINX.PRG  

*: Calls: PAR_E.PRG  

*:       : PAR_F.PRG  

*:       : PAR_G.PRG  

*:       : PAR_H.PRG  

*:       : PAR_I.PRG  

*:       : PAR_K.PRG  

*:       : PAR_L.PRG  

*:       : REPM_INT.PRG  

*:       : PAR_O.PRG  

*:       : NTFST.PRG  

*:       : CERTIFY      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:38           SNAP! version 3.12e
*:*****  

*  

* Program sets parameters for all batched reports  

*
*
```

```

PRIVATE nn,temp,tchr,flag
nn = 0
reportch = 'A'

DO WHILE nn < 17
  tchr = CHR(ASC('A')+nn)
  option = 1
  IF tchr$reportst .OR. tchr = "Q"
    flag = 0
    DO CASE
      CASE tchr = "E"
        DO par_e
        CLEAR
        @ 10,10 SAY "Report E -- PSSR/Schedule/Submittal Comparison 1"
        @ 11,10 SAY "will be run with:"
        @ 13,20 SAY "FLOAT = "+RTRIM(LTRIM(STR(refloat)))+" working days."
        flag = 1
      CASE tchr = "F"
        DO par_f
        CLEAR

```

```

@ 10,10 SAY "Report F -- PSSR/Schedule/Submittal Comparison 2"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "% FLOAT = "+LTRIM(RTRIM(STR(rfppercent)))
flag = 1
CASE tchr = "G"
DO par_g
CLEAR
@ 10,10 SAY "Report G -- Resource Loading Analysis - Manpower Variance"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "Men/Day      = "+LTRIM(RTRIM(STR(rgmn)))
@ 14,20 SAY "STD.DEV.      = "+LTRIM(RTRIM(STR(rgsd)))+" for the threshold"
@ 15,20 SAY "Grouping Code = "+LTRIM(RTRIM(rgcode))
@ 17,30 SAY "1 = STND 2 = RESP 3 = WAD      4 = SORT"
@ 18,30 SAY "5 = SUB   6 = AREA 7 = SUB-AREA 8 = TYPE"
flag = 1
CASE tchr = "H"
DO par_h
CLEAR
@ 10,10 SAY "Report H -- Resource Loading Analysis - Funding Variance"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "$/Day      = "+LTRIM(RTRIM(STR(rhdl)))+"."
@ 14,20 SAY "STD.DEV.      = "+LTRIM(RTRIM(STR(rhsd)))+" for the threshold"
@ 15,20 SAY "Grouping Code = "+LTRIM(RTRIM(rhcode))
@ 17,30 SAY "1 = STND 2 = RESP 3 = WAD      4 = SORT"
@ 18,30 SAY "5 = SUB   6 = AREA 7 = SUB-AREA 8 = TYPE"
flag = 1
CASE tchr = "I"
DO par_i
CLEAR
@ 10,10 SAY "Report I -- Resource Loading Analysis - Time Variance"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "Days/activity = "+LTRIM(RTRIM(STR(ridy)))+"."
@ 14,20 SAY "STD.DEV.      = "+LTRIM(RTRIM(STR(risd)))+" for the threshold"
@ 15,20 SAY "Grouping Code = "+LTRIM(RTRIM(ricode))
@ 17,30 SAY "1 = STND 2 = RESP 3 = WAD      4 = SORT"
@ 18,30 SAY "5 = SUB   6 = AREA 7 = SUB-AREA 8 = TYPE"
flag = 1
CASE tchr = "K"
DO par_k
CLEAR
@ 10,10 SAY "Report K -- Activities to Crash"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "Man Power Loading    = "+LTRIM(RTRIM(STR(mp1)))+"."
@ 14,20 SAY "Duration          = "+LTRIM(RTRIM(STR(durat)))
@ 15,20 SAY "Relational Duration = "+LTRIM(RTRIM(STR(reldur)))+"%"
flag = 1
CASE tchr = "L"
DO par_l
CLEAR
@ 10,10 SAY "Report L -- Delays in Design"
@ 11,10 SAY "will be run with:"
@ 13,20 SAY "Previous schedule = "+LTRIM(TRIM(filsch1))

```

```

@ 14,20 SAY "Current schedule = "+LTRIM(TRIM(filsch21))
@ 15,20 SAY "STND Code      = "+LTRIM(RTRIM(stnd))
@ 16,20 SAY "Float        = "+LTRIM(RTRIM(STR(ufloat)))
@ 17,20 SAY "% Difference Limit = "+LTRIM(RTRIM(STR(upercent*100)))+"%"
flag = 1
CASE tchr = "M"
  DO repm_int
  CLEAR
  IF option = 0
    RETURN
  ENDIF
  @ 10,10 SAY "Report M -- Contractor Grading"
  @ 11,10 SAY "will be run with:"
  @ 13,20 SAY "Data Date = "+DTOC(datadate)
  @ 15,20 SAY "Grading scales as entered or accepted"
  IF mch = 'S'
    @ 17,20 SAY "Summary reports will be run."
  ELSE
    IF mch = 'D'
      @ 17,20 SAY "Detail reports will be run."
    ELSE
      @ 17,20 SAY "Summary and Detail reports will be run."
    ENDIF
  ENDIF
  flag = 1
CASE tchr = "O"
  DO par_o
  CLEAR
  @ 10,10 SAY "Report O -- Slippage"
  @ 11,10 SAY "will be run with:"
  @ 13,20 SAY "Schedule 1 (most recent) = "+TRIM(filsched1)
  @ 14,20 SAY "Schedule 2          = "+TRIM(filsched2)
  @ 15,20 SAY "Schedule 3          = "+TRIM(filsched3)
  @ 16,20 SAY "Schedule 4          = "+TRIM(filsched4)
  @ 17,20 SAY "Schedule 5 (oldest)  = "+TRIM(filsched5)
  flag = 1
CASE tchr = "Q"
  IF 'R'$reportst.OR.'S'$reportst.OR.'T'$reportst.OR.'U'$reportst.OR. ;
    'V'$reportst.OR.'W'$reportst.OR.'X'$reportst
    DO ntfst
    CLEAR
    @ 10,10 SAY "Report R through X"
    @ 11,10 SAY "will be run with:"
    @ 13,20 SAY "Recent Schedule   = "+TRIM(ntfsched1)
    @ 14,20 SAY "Recent Logic     = "+TRIM(ntflogic1)
    @ 15,20 SAY "Previous Schedule = "+TRIM(ntfsched2)
    @ 16,20 SAY "Previous Logic   = "+TRIM(ntflogic2)
    IF option = 0
      RETURN
    ENDIF
  ENDIF
  flag = 1

```

```
ENDCASE
IF flag = 1
  DO certify
    flag = 0
  ENDIF
ENDIF
IF option = 0
  RETURN
ELSE
  IF option < 8
    nn = nn + 1
    option = 1
  ENDIF
ENDIF
ENDDO
*: EOF: SETPARMS.PRG
^Z
```

```
*:*****  
*: Program: KEYPRESS.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 03/28/89 22:10  
*: Called by: MAINX.PRG  
*: Calls: QUIT.PRG  
*: Documented 04/27/89 at 10:38 SNAP! version 3.12e  
*:*****  
* KEYPRESS.PRG - What to do if user presses a key during batch operation  
dummy=INKEY() && Clear out keypress  
zz=.n.  
*SET COLO TO GR+/R,GR+/R*  
Q24,60 SAY 'Abort (y/N)?' GET zz PICTURE 'Y'  
*SET COLO TO  
READ  
IF zz  
DO QUIT  
ENDIF  
Q24,60  
RETURN  
*: EOF: KEYPRESS.PRG  
^Z
```

```

*:*****  

*: Program: INTRO.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 17:44  

*: Called by: MAINX.PRG  

*: Calls: ?SPACE() (function in ?)  

*: : .NOT.EOF() (function in ?)  

*: : .AND..NOT.EOF() (function in ?)  

*: Documented 04/27/89 at 10:39 SNAP! version 3.12e  

*:*****  

** INTRO.PRG - Get user's choice of output device; print intro to each Report  

SET COLOR TO &outputcol  

@24,0  

CLEAR  

* Do Intro for each Report  

IF TRIM(reportch) > TRIM('Q')  

    RETURN  

ENDIF  

?&intro_on  

?  

?SPACE(17)+'PROJECT MANAGEMENT TOOLS: REPORT '+reportch  

?  

?  

DO CASE  

CASE reportch='A'  

    SET COLOR TO &extracol  

    @24,4 SAY 'REPORT A: PSSR/Schedule Matching'  

    SET COLOR TO &outputcol  

    ?SPACE(23)+'PSSR/SCHEDULE MATCHING'  

?  

?&intro_off  

?{      ALGORITHM: Report A prints data for each Schedule Activity which matches one or more  

PSSR (Procurement)  

?{      File) Activity ID numbers. (Note that duplicate Activity ID numbers in the PSSR files  

are repeated if)  

?{      they match more than 1 Schedule Activity ID number, using a match of the first 9  

characters.)  

?  

?{      USER-MODIFIABLE ITEMS: Initial PSSR and Schedule filenames.]  

?  

?{      OUTPUT: Output from Report A follows the same format as that from Reports C, D, E and  

F. The following)  

?{      items are listed from each matching Schedule Activity, in this order: Schedule Activity  

ID number, Title;  

?{      (in text), Early Start date, Early Finish date, Late Start date, Late Finish date, Float

```

```

(in days), Original]
? [ Duration of Activity (in days), Remaining Duration of Activity (in days, and at the time
the Schedule was]
? [ last calculated in Primevera), and Percent Completed of Activity.]
?
? [ For all PSSR Activities matching each Schedule Activity, the following items are listed
in this order:]
? [ PSSR Activity ID number, Description (text), Vendor Name (text), Arrive on Site date,
Required date, the]
? [ difference in days between the Schedule Early Start date and the PSSR Arrive on Site
date, and an arrow for]
? [ each group of matching PSSR Activities indicating which of them shows the largest
difference between the]
? [ Schedule Early Start date and the PSSR Arrive on Site date.]
?
? [ At the end of Report A, the number of Schedule Activities and the number of PSSR
Activities which met]
? [ the search criteria is listed.]


CASE reportch='B'
SET COLOR TO &extracol
Q24,4 SAY "REPORT B: PSSR/Schedule 'PR' Code Matching"
SET COLOR TO &outputcol
?SPACE(19)+[PSSR/SCHEUDLE 'PR' CODE MATCHING]
?
?&intro_off
? [ ALGORJTHM: Report B prints data for each Schedule Activity which is a 'Procurement'
Activity (and thus)
? [ has a 'PR' or a 'PR-' embedded in its Activity Number), AND lacks any matching Activities
in the the PSSR]
? [ (Procurement) File. Therefore, Report B only lists Activity data from the Schedule
files.]
?
? [ USER-MODIFIABLE ITEMS: Initial PSSR and Schedule filenames.]
?
? [ OUTPUT: Output from Report B includes the following items listed from each Schedule
Activity meeting the]
? [ above criteria, in this order: Schedule Activity ID number, Title (in text), Early
Start date, Early Finish]
? [ date, Late Start date, Late Finish date, Float (in days), Original Duration of Activity
(in days), Remaining]
? [ Duration of Activity (in days, and at the time the Schedule was last calculated in
Primevera), and Percent]
? [ Completed of Activity.]
?
? [ At the end of Report B, the number of Schedule Activities which met the search criteria
is listed.]


CASE reportch='C'
SET COLOR TO &extracol
Q24,4 SAY 'REPORT C: PSSR/Schedule; Date Comparison 1'
SET COLOR TO &outputcol

```

```

?SPACE(19) +'PSSR/SCHEDULE; DATE COMPARISON 1'
?
?&intro_off
?[      ALGORITHM: Report C prints data for each Schedule Activity which matches one or more
PSSR)
?{      (Procurement File) Activity ID numbers, AND for which the latest PSSR Arrive at Site
date is later than the]
?[      matching Schedule Early Start date, AND for which the matching Schedule Activity has
a Float of less than 10]
?[      working days. (Note that duplicate Activity ID numbers in the PSSR files are repeated
if they match more)
?[      than 1 Schedule Activity ID number, using a match of the first 9 characters.)
?
?{      USER-MODIFIABLE ITEMS: Initial PSSR and Schedule filenames.]
?
?{      OUTPUT: Output from Report C follows the same format as that from Reports A, D, E and
F. The following]
?[      items are listed from each matching Schedule Activity, in this order: Schedule Activity
ID number, Title]
?[      (in text), Early Start date, Early Finish date, Late Start date, Late Finish date, Float
(in days), Original]
?[      Duration, of Activity (in days), Remaining Duration of Activity (in days, and at the
time the Schedule was]
?[      last calculated in Primavera), and Percent Completed of Activity.]
?
?{      For all PSSR Activities matching each Schedule Activity, the following items are listed
in this order: ]
?[      PSSR Activity ID number, Description (text), Vendor Name (text), Arrive on Site date,
Required date, the]
?[      difference in days between the Schedule Early Start date and the PSSR Arrive on Site
date, and an arrow for]
?[      each group of matching PSSR Activities indicating which of them shows the largest
difference between the]
?[      Schedule Early Start date and the PSSR Arrive on Site date.]
?
?{      At the end of Report C, the number of Schedule Activities and the number of PSSR
Activities which met]
?[      the search criteria is listed.]

```

```

CASE reportch='D'
SET COLOR TO &extracol
@24,4 SAY 'REPORT D: PSSR/Schedule; Date Comparison 2'
SET COLOR TO &outputcol
?SPACE(19) +'PSSR/SCHEDULE; DATE COMPARISON 2'
?
?&intro_off
?[      ALGORITHM: Report D prints data for each Schedule Activity which matches one or more
PSSR (Procurement)
?[      File) Activity ID numbers, AND for which the latest PSSR Arrive at Site date is later
than the matching]
?[      Schedule Early Start date by a number of days greater than 20% of the Float for that
Activity, AND for]

```

?[ which the matching Schedule Activity had a Float of greater than 10 working days. (Note  
 that duplicate)  
 ?[ Activity ID numbers in the PSSR files are repeated if they match more than 1 Schedule  
 Activity ID number.]  
 ?[ using a match of the first 9 characters.)]  
 ?  
 ?[ USER-MODIFIABLE ITEMS: Initial PSSR and Schedule filenames.]  
 ?  
 ?[ OUTPUT: Output from Report D follows the same format as that from Reports A, C, E and  
 F. The following]  
 ?[ items are listed from each matching Schedule Activity, in this order: Schedule Activity  
 ID number, Title]  
 ?[ (in text), Early Start date, Early Finish date, Late Start date, Late Finish date, Float  
 (in days), Original]  
 ?[ Duration of Activity (in days), Remaining Duration of Activity (in days, and at the time  
 the Schedule was]  
 ?[ last calculated in Primavera), and Percent Completed of Activity.]  
 ?  
 ?[ For all PSSR Activities matching each Schedule Activity, the following items are listed  
 in this order:  
 ?[ PSSR Activity ID number, Description (text), Vendor Name (text), Arrive on Site date,  
 Required date, the]  
 ?[ difference in days between the Schedule Early Start date and the PSSR Arrive on Site  
 date, and an arrow for]  
 ?[ each group of matching PSSR Activities indicating which of them shows the largest  
 difference between the]  
 ?[ Schedule Early Start date and the PSSR Arrive on Site date.]  
 ?  
 ?[ At the end of Report D, the number of Schedule Activities and the number of PSSR  
 Activities which met]  
 ?[ the search criteria is listed.]

CASE reportch='E'  
 SET COLOR TO &extracol  
 @24,4 SAY 'REPORT E: PSSR/Schedule/Submittal Comparison 1'  
 SET COLOR TO &outputcol  
 ?SPACE(17)+'PSS /SCHEDULE/SUBMITTAL COMPARISON 1'  
 ?  
 ?&intro\_off  
 ?[ ALGORITHM: Report E compares date consistency among Activities having matching  
 information in the Schedule.]  
 ?[ PSSR, and Submittal Register files. Data is printed for each Schedule Activity which  
 matches one or more PSSR]  
 ?[ (Procurement) File Activities AND contains matching information in the Submittal file,  
 AND for which the PSSR]  
 ?[ 'Submittal to CE' and 'CE Submittal Approval' dates do not match, AND for which the  
 Schedule Activity has less]  
 ?[ than (by default) 10 days of Float. (Note that duplicate Activity ID numbers in the  
 PSSR files are repeated)]  
 ?[ if they match more than 1 Schedule Activity ID number, using a match of the first 9  
 characters.)]  
 ?

?[       USER-MODIFIABLE ITEMS: Initial PSSR, Schedule and Submittal filenames. Also, the user can change the value]

?[       of the float (10 days by default).]

?'       FOR THIS RUN, THESE VALUES ARE: Float = '+LTRIM(STR(refloat))+'' days'

?

?[       OUTPUT: Output from Report E follows the same format as that from Reports A, C, D and F. The following items]

?[       are listed from each matching Schedule Activity, in this order: Schedule Activity ID number, Title (in text),.]

?[       Early Start date, Early Finish date, Late Start date, Late Finish date, Float (in days), Original Duration]

?[       of activity (in days), Remaining Duration of Activity (in days, and at the time the Schedule was last)

?[       calculated in Primavera), and Percent Completed of Activity.]

?

?[       For all PSSR Activities matching each Schedule Activity, the following items are listed in this order:]

?[       PSSR Activity ID number, Description (text), Vendor Name (text), Arrive on Site date, Required date, the]

?[       difference in days between the Schedule Early Start date and the PSSR Arrive on Site date, and an arrow for]

?[       each group of matching PSSR Activities indicating which of them shows the largest difference between the]

?[       Schedule Early Start date and the PSSR Arrive on Site date.]

?

?[       At the end of Report E, the number of Schedule Activities and the number of PSSR Activities which met]

?[       the search criteria is listed.]

CASE reportch='F'

SET COLOR TO &extraccl

@24,4 SAY 'REPORT F: PSSR/Schedule/Submittal Comparison 2'

SET COLOR TO &outputcol

?SPACE(17) +'PSSR/SCHEDULE/SUBMITTAL COMPARISON 2'

?

?&intro\_off

?[       ALGORITHM: Report F compares date consistency among Activities having matching information in the Schedule,]

?[       PSSR, and Submittal Register files. Data is printed for each Schedule Activity which matches one or more PSSR]

?[       (Procurement) File Activities AND contains matching information in the Submittal file, AND for which the PSSR]

?[       'Submittal to CE' and 'CE Submittal Approval' dates do not match, AND for which the difference between the PSSR]

?[       dates and the Schedule dates is greater than (by default) 20% of the Float. (Note that duplicate Activity ID)

?[       numbers in the PSSR files are repeated if they match more than 1 Schedule Activity ID number, using a match of]

?[       the first 9 characters.)]

?

?[       USER-MODIFIABLE ITEMS: Initial PSSR, Schedule and Submittal filenames. Also, the user can change the percentage]

```

?{      difference (20% by default), as defined above.]  

?'      FOR THIS RUN, THESE VALUES ARE: Percentage = '+LTRIM(STR(rfpercent))+' %'  

?  

?{      OUTPUT: Output from Report F follows the same format as that from Reports A, C, D and  

E. The following items]  

?{      are listed from each matching Schedule Activity, in this order: Schedule Activity ID  

number, Title (in text).]  

?{      Early Start date, Early Finish date, Late Start date, Late Finish date, Float (in days),  

Original Duration]  

?{      of Activity (in days), Remaining Duration of Activity (in days, and at the time the  

Schedule was last]  

?{      calculated in Primavera), and Percent Completed of Activity.]  

?  

?{      For all PSSR Activities matching each Schedule Activity, the following items are listed  

in this order:]  

?{      PSSR Activity ID number, Description (text), Vendor Name (text), Arrive on Site date,  

Required date, the]  

?{      difference in days between the Schedule Early Start date and the PSSR Arrive on Site  

date, and an arrow for]  

?{      each group of matching PSSR Activities indicating which of them shows the largest  

difference between the]  

?{      Schedule Early Start date and the PSSR Arrive on Site date.]  

?  

?{      At the end of Report F, the number of Schedule Activities and the number of PSSR  

Activities which met]  

?{      the search criteria is listed.]
```

```

CASE reportch='G'      && Resource Loading Analysis Module/Manpower Variance  

SET COLOR TO &extracol  

Q24,4 SAY 'REPORT G: Resource Loading Analysis; Manpower Variance'  

SET COLOR TO &outputcol  

?SPACE(13)+'RESOURCE LOADING ANALYSIS: MANPOWER VARIANCE'  

?  

?&intro_off  

?{      ALGORITHM: Report G selects and prints those Activities requiring more than (by  

default) 20 Men Per Day, and]  

?{      calculates and prints the Number in the Group, the Mean and Standard Deviation for the  

Men Per Day for groups of]  

?{      Activities in the same (by default) CSI Standard Code group. It identifies those  

Activities for which the Men Per]  

?{      Day is greater than (by default) 1.00 Standard Deviations above OR below the group Mean,  

and lists the number of]  

?{      multiples of the Men Per Day value, relative to the group Mean. (Activities with "Pct"  

= 100% are screened out.)}  

?  

?{      USER-MODIFIABLE ITEMS: Initial Schedule filename. Also, the user can change the value  

of the number of]  

?{      Men Per Day per Activity above which to search (20 men by default), the threshold number  

of Standard Deviations]  

?{      1.00 by default) above or below the group mean by which to flag particular Activities,  

and the Activity Code or]  

?{      Activity ID Code (the STND Code by default) used to group Activities. All of these
```

user-defined items are]

?[ listed on this title page; in addition, the Activity Code or Activity ID Code and the threshold number of]

?[ Standard Deviations are included in the header of each page.]

?' FOR THIS RUN, THESE VALUES ARE: Men per Day = '+LTRIM(STR(rgmn))+'; Standard Deviation Multiples = '+LTRIM(STR(rgsd,10,2))+'; Code = '+code&rgcode

?

?[ OUTPUT: Output from Report G includes the following items listed from each Schedule Activity meeting the]

?[ above criteria, in this order: Schedule Activity ID number, Title (in text), Early Start date, Early Finish]

?[ date, Late Start date, Late Finish date, Float (in days), Original Duration of Activity (in days), Remaining]

?[ Duration of Activity (in days, and at the time the Schedule was last calculated in Primevera), and Percent]

?[ Completed of Activity. Also output are the number of Men Per Day required by that Activity, and a number]

?[ indicating the relative difference from the group Mean of the Men Per Day value of that Activity.]

?

?[ At the end of Report G, the number of matching Schedule Activities found is listed.]

CASE reportch='H' &

SET COLOR TO &extracol

@24,4 SAY 'REPORT H: Resource Loading Analysis; Cost Variance'

SET COLOR TO &outputcol

?SPACE(15) +'RESOURCE LOADING ANALYSIS: COST VARIANCE'

?

?&intro\_off

?[ ALGORITHM: Report H selects and prints those Activities requiring more than (by default) 2,000.00 Dollars Per Day, and]

?[ calculates and prints the Number in the Group, the Mean and Standard Deviation for the Dollars Per Day for groups of]

?[ Activities in the same (by default) CSI Standard Code group. It identifies those Activities for which the Dollars Per]

?[ Day is greater than (by default) 1.00 Standard Deviations above OR below the group Mean, and lists the number of]

?[ multiples of the Dollars Per Day value, relative to the group Mean. (Activities with "Pct" = 100% are screened out.)]

?

?[ USER-MODIFIABLE ITEMS: Initial Schedule filename. Also, the user can change the value of the number of]

?[ Dollars Per Day per Activity above which to search (\$2000 by default), the threshold number of Standard Deviations]

?[ 1.00 by default) above or below the group mean by which to flag particular Activities, and the Activity Code or]

?[ Activity ID Code (the STND Code by default) used to group Activities. All of these user-defined items are]

?[ listed on this title page; in addition, the Activity Code or Activity ID Code and the threshold number of]

?[ Standard Deviations are included in the header of each page.]

?' FOR THIS RUN, THESE VALUES ARE: Dollars Per Day = '+LTRIM(STR(rhd1))+'' Standard

```

Deviation Multiples = '+LTRIM(STR(rhsd,10,2))+'; code = '+code&rhcode
?
?{      OUTPUT: Output from Report H includes the following items listed from each Schedule
Activity meeting the]
?{      above criteria, in this order: Schedule Activity ID number, Title (in text), Early
Start date, Early Finish]
?{      date, Late Start date, Late Finish date, Float (in days), Original Duration of Activity
(in days), Remaining]
?{      Duration of Activity (in days, and at the time the Schedule was last calculated in
Primavera), and Percent]
?{      Completed of Activity. Also output are the number of Dollars Per Day required by that
Activity, and a number]
?{      indicating the relative difference from the group Mean of the Dollars Per Day value of
that Activity.]
?
?{      At the end of Report H, the number of matching Schedule Activities found is listed.}

CASE reportch='I'      ;;
SET COLOR TO $extracol
@24,4 SAY 'REPORT I: Resource Loading Analysis; Time Variance'
SET COLOR TO $outputcol
?SPACE(14) +'PPSOURCE ANALYSIS: TIME VARIANCE'
?
?&in:ro_off
?{      ALGORITHM: Report I selects and prints those Activities requiring more than (by
default) 40 Working Days, and]
?{      calculates and prints the Number in the Group, the Mean and Standard Deviation for the
Working Days for groups of]
?{      Activities in the same (by default) CSI Standard Code group. It identifies those
Activities for which the Working]
?{      Days is greater than (by default) 1.00 Standard Deviations above OR below the group
Mean, and lists the number of]
?{      multiples of the Working Days value, relative to the group Mean. (Activities with "Pct"
= 100% are screened out.)]
?
?{      USER-MODIFIABLE ITEMS: Initial Schedule filename. Also, the user can change the value
of the number of]
?{      Working Days per Activity above which to search (40 Days by default), the threshold
number of Standard Deviations]
?{      1.00 by default) above or below the group mean by which to flag particular Activities,
and the Activity Code or]
?{      Activity ID Code (the STND Code by default) used to group Activities. All of these
user-defined items are]
?{      listed on this title page; in addition, the Activity Code or Activity ID Code and the
threshold number of]
?{      Standard Deviations are included in the header of each page.]
?'
FOR THIS RUN, THESE VALUES ARE: Working Days = '+LTRIM(STR(ridy))+'
Standard Deviation Multiples = '+LTRIM(STR(risd,10,2))+'; Code = '+code&ricode
?
?{      OUTPUT: Output from Report I includes the following items listed from each Schedule
Activity meeting the]
?{      above criteria, in this order: Schedule Activity ID number, Title (in text), Early

```

```

Start date, Early Finish]
? [      date, Late Start date, Late Finish date, Float (in days), Original Duration of Activity
(in days), Remaining]
? [      Duration of Activity (in days, and at the time the Schedule was last calculated in
Primavera), and Percent]
? [      Completed of Activity. Also output are the number of Working Days required by that
Activity, and a number]
? [      indicating the relative difference from the group Mean of the Working Days value of that
Activity.]
?
? [      At the end of Report I, the number of matching Schedule Activities found is listed.]


CASE reportch='J'      && Logical Analysis Module'
SET COLOR TO $extracol
@24,4 SAY 'REPORT J: Logical Analysis'
SET COLOR TO $outputcol
?SPACE(27)+'LOGICAL ANALYSIS'
?

?&intro_off
? [      ALGORITHM: Report J identifies and prints the construction and other Activities on
each floor which do not follow]
? [      a normal construction progression sequence in their "Early Start Dates". This sequence
is defined by examining (by)
? [          default) the "Sort Details" Code and "Early Start Date" for each Activity in the
Schedule file. The normal progressions]
? [          are explained in detail on the following page - Activities deviating from this
progression are identified and printed.]
? [          In addition, any drywall Activities (identified as such by their Sort Details Codes)
which began before roofing]
? [          Activities are started, are identified and printed.]
?
? [          USER-MODIFIABLE ITEMS: Initial Schedule filename.]
?
?'
FOR THIS RUN, THIS VALUE IS: Code Used = '+rjcode+' : '+code&rjcode
'

? [      OUTPUT: Output from Report J includes the following items listed from each matching
Schedule Activity, in]
? [          this order: Schedule Activity ID number, Title (in text), Early Start date, Early
Finish date, Late Start]
? [          date, Late Finish date, Float (in days), Original Duration of Activity (in days),
Remaining Duration of]
? [          Activity (in days, and at the time the Schedule was last calculated in Primavera),
Percent Completed of]
? [          of Activity, and the Sort Detail Code and Definition. Also, the Sort Detail Code of
the Activity on that floor]
? [          which immediately prior (according to the "Early Start Date") to this Activity, is
printed as an indicator of]
? [          how much "out of sequence" this Activity is.]


CASE reportch='K'      && Activities to Crash Module
SET COLOR TO $extracol
@24,4 SAY 'REPORT K: Activities to Crash'

```

```

SET COLOR TO &outputcol
?SPACE(24)+'ACTIVITIES TO CRASH'
?
?&intro_off
?{           ALGORITHM: Report K prints the four most critical paths in the schedule database.
It does this by ]
?{           sorting schedule information by total float and finish date and then searching for the
four most]
?{           critical paths. Report K also analyzes the Man Power Loading for Critical Path
Activities.]
?{           The report lists those activities whose Man Power Loading is greater than a user
specified amount.]
?{           Activities with durations greater than a user specified amount (default = 20) are also
listed.]
?{           The final section of report K identifies activities with relation Durations in excess
of a user]
?{           specified percentage (default = 20%).]
?

?{           USER-MODIFIABLE ITEMS: Man Power Loading Value, Duration Value, and Duration Relation
Percentage. ]
?

?{           OUTPUT: Output from Report K includes the following for each critical path. ]
?{           Schedule Activity ID number, Title, Early Start Date, Early Finish Date, Late Start
Date, Late Finish]
?{           Date, Actual Start Date, Actual Finish Date, Float, remaining Duration, percent
complete, original]
?{           duration, lag and relation. Output for Manpower loading, remaining duration and
relation ]
?{           durations follow each critical path.]
?

CASE reportch='L'      && Delays in Design Module
SET COLOR TO &extracol
Q24,4 SAY 'REPORT L: Delays in Design'
SET COLOR TO &outputcol
?SPACE(22)+'DELAYS IN DESIGN'
?
?&intro_off
?{           ALGORITHM: Report L requires data from two Schedule update files. It prints data for
each Activity which]
?{           has a (by default) "DR" STND code, AND which has a Float greater than (by default) 10
days, AND which has lost]
?{           (by default) 10% of that Float from one Schedule update to the next.]
?
?{           USER-MODIFIABLE ITEMS: Initial Schedule update filenames. Also, the user can change
the value of the float]
?{           (10 days by default) and the percentage difference (20% by default).]
?'
FOR THIS RUN, THESE VALUES ARE.  Float = '+LTRIM(STR(ufloat))+'' days; Percentage =
'+LTRIM(STR(upercent*100))+'' %
?
?{           OUTPUT: The following items are listed from each matching Schedule Activity, in this
order: Schedule]

```

```

?{           Activity ID number, Title (in text), Early Start date, Early Finish date, Late Start
date, Late Finish]
?{           date, Float (in days), Original Duration of Activity (in days), Remaining Duration of
Activity (in days.)
?{           and at the time the Schedule was last calculated in Primavera), and Percent Completed
of Activity.}

CASE reportch='M'      && Contractor Grading Module'
SET COLOR TO &extracol
Q24,4 SAY 'REPORT M: Contractor Grading'
SET COLOR TO &outputcol
?SPACE(25)+'CONTRACTOR GRADING'
?
?<>intro_off
?{           ALGORITHM: Report M requires data from 1 Schedule update file, and a matching PSSR
and Submittal file.]
?{           First, it calculates a "rating" of the percentage of PSSR and of Submittal Activities
which correspond to]
?{           Schedule Activities, the percentage of PSSR and of Submittal Activities which correspond
to Schedule procurement]
?{           Activities, the number of non-procurement Activities with durations exceeding 20 days,
and the number of]
?{           correctly coded Schedule Activities (assuming that the code definitions in the "NTF"
Codes Dictionary within]
?{           Primavera are themselves correct.)}
?
?{           Also, it sums the number of Schedule Activities having negative float, determines the
percentage of]
?{           Activities on the four most critical paths which met their "Early Start Date", analyzes
the Activities on the]
?{           four most critical paths in terms of the time remaining and the time expended, analyzes
the percentage of]
?{           Activities which have adequate "Actual Start Dates", analyzes the percentage of
Activities having a "Percent"]
?{           Complete" greater than 0 or a "Remaining Duration" with no "Actual Start Date", and
analyzes the percentage of]
?{           Activities with "Remaining Durations" greater than their "Original Durations".}
?
?{           Note that the user must run "Report K: Activities to Crash" before running this Report,
in order to determine]
?{           the four most critical paths in the Schedule file.}
?
?{           USER-MODIFIABLE ITEMS: Initial Schedule update filenames. Also, the user can change
all of the grading]
?{           "threshold" values, which are used to determine the "grade" or rating given to the
contractor. The user]
?{           must input the correct Primavera Data Date in order for one of the subitems to work,
since the "Data Date"]
?{           cannot be included in Primavera Dbase export files.) The user can choose whether to
output the Detailed]
?{           Report, the Summary Report, or both.}
?

```

```

? [      OUTPUT: 1) A Summary Report, which gives the grade for each subitem in the two overall
grading items, in]
? [      English-like text, and/or 2) A Detail Report, which lists many data items concerning
the Activities matched]
? [      in the Report M algorithms and summarized in the Summary Report.]
CASE reportch='O'      && Activity Slippage
  SET COLOR TO &extracol
  Q24,4 SAY 'REPORT O: Activity Slippage'
  SET COLOR TO &outputcol
  ?SPACE(26)+'ACTIVITY SLIPPAGE'
?
?&intro_off
? [      ALGORITHM: Report O accepts up to five schedule files and compares an activity's early
start date, duration,]
? [      float, and productivity (defined as actual duration/original duration). If there is
a change from the previous]
? [      schedule file, the activity is printed. If the productivity index exceeds 1 standard
deviation, the number of]
? [      standard deviations that the productivity rate is over, is printed.]
?
? [      USER-MODIFIABLE ITEMS: The user inputs up to five schedule files which are checked
to make sure that they]
? [      are valid. The user may also enter which code to sort on. The default is STND]
?
? [      OUTPUT: The following items are listed from each matching Schedule Activity, in this
order: Schedule]
? [      Activity ID number, Title (in text), Early Start date, Early Finish date, Late Start
date, Late Finish]
? [      date, Float (in days), Original Duration of Activity (in days), Remaining Duration of
Activity (in days,]
? [      and at the time the Schedule was last calculated in Primavera), and Percent Completed
of Activity, the]
? [      Productivity Rate, and, if applicable, the number of standard deviations over the mean.]
ENDCASE
* Now print intro filenames, no. of Activities in each file, etc.
?
?
?SPACE(9)+'THIS REPORT WAS RUN ON '+DTOC(DATE())+' AT '+TIME()
IF reportch='A'.OR.reportch='B'.OR.reportch='C'.OR.reportch='D'.OR.reportch='E'.OR.reportch='F'

SELECT B
GO TOP
N=0
DO WHILE .NOT.EOF()
  X=TRIM(activ_id)
  SKIP
  IF X=TRIM(activ_id).AND..NOT.EOF()
    N=N+1
  ENDIF

```

```
ENDDO
?SPACE(9) +'YOUR INITIAL PSSR FILE(S), '+LTRIM(filpssr_pr)+' AND '+LTRIM(filpssr_sc)+', CONTAIN
'+LTRIM(STR(RECCOUNT()))+' ACTIVITY ID NUMBER RECORDS, OF WHICH '+LTRIM(STR(N))+ 'ARE DUPLICATES.'
ENDIF
SELECT D
?SPACE(9) +'YOUR INITIAL PROJECT SCHEDULE FILE(S), '+LTRIM(filsched)
?SPACE(9) +'AND '+LTRIM(filsucces)+', CONTAIN '+LTRIM(STR(RECCOUNT()))+' ACTIVITY ID NUMBER RECORDS.'
IF reportch='E'.OR.reportch='F'
  SELECT E
  ?SPACE(9) +'YOUR INITIAL SUBMITTAL FILE, '+LTRIM(filsubmi)+', CONTAINS '+LTRIM(STR(RECCOUNT()))+
ACTIVITY ID NUMBER RECORDS.'
ENDIF
*?eop
RETURN
*: EOF: INTRO.PRG
^Z
```

```

*:*****  

*: Program: REPA.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      23:53  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e
*:*****  

** REPA.PRG - Report A, TASK 1: PSSR/Schedule Matching  

* Mark McReynolds, 1/11/89  

SELECT D  && schedule.dbf  

GO TOP  

STORE 0 TO curline,page,smatches,pmatches  

area=' '    && AREA Code  

stnd=' '    && STND Code  

header2=.F.  

SET EXACT OFF  

DO WHILE .NOT. EOF()  

  SELECT B  

  GO TOP  

  SEEK SUBSTR(d->act,1,9) && Substring must stay at Act,1,9 to get all matches  

  IF FOUND()  

    IF curline>maxlinesa  

      DO header1  

    ENDIF  

    IF area <> SUBSTR(d->act,1,1) .AND. area <> '4'  

      area=SUBSTR(d->act,1,1)  

      DO header1  

    ENDIF  

    IF stnd <> SUBSTR(d->act,5,2)  

      stnd=SUBSTR(d->act,5,2)  

      ? ' ****' +stnd+ ' : ' +TRIM(stndv&stnd)+ ' *****';  

      curline=curline+1  

    ENDIF  

    ? SPACE(7)+d->act+ ' +d->title+ ' +DTOC(d->es)+ ' +DTOC(d->ef)+ ' +DTOC(d->ls);  

    +' '+DTOC(d->lf)+ ' +STR(d->tf,7)+ ' +STR(d->od,7)+STR(d->rd,6)+;  

    STR(d->pct,6)  

    smatches=smatches+1  

    curline=curline+1  

    IF curline>maxlinesa  

      DO header1  

    ENDIF  

    area=SUBSTR(d->act,1,1)  

    IF header2  

      ? SPACE(31)+'PSSR

```

VENDOR

```

ARRIVE
? SPACE(31)+'ACTIVITY ID      DESCRIPTION (from PSSR files)      NAME
    ON SITE      REQUIRED'
?   SPACE(31)+-----+
-----+
?
curline=curline+4
header2=.n.
ENDIF
start=RECNO()
N=1
big=CTOD(arrivesite)
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
  IF CTOD(arrivesite)>big
    big=CTOD(arrivesite)
  ENDIF
  N=N+1
  SKIP
ENDDO
GOTO start
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
  ? SPACE(31)+activ_id+'      '+descript+'      '+vendorname+'      '+arrivesite+;
  '      '+required+'      '
curline=curline+1
pmatches=pmatches+1
IF curline>maxlinesa
  DO header1
ENDIF
SKIP
ENDDO
?
curline=curline+1
ENDIF
SELECT D
IF EOF()
  EXIT
ENDIF
SKIP
ENDDO
? eop
?
? &footer_on
? SPACE(50)+'*** END OF REPORT A ***'
? '      *** THERE WERE '+LTRIM(STR(smatches))+ ' SCHEDULE ACTIVITY ID NUMBERS, WHICH MATCHED
'+LTRIM(STR(pmatches))+ ' PSSR ACTIVITY ID NUMBERS, INCLUDING PSSR MULTIPLES. ***'
? &footer_off
? eop
RETURN
*: EOF: REPA.PRG
^Z

```

```

*:*****  

*: Program: REPB.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      23:51  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPB.PRG - Report B, TASK 1: PSSR/Schedule 'PR' Code Matching  

* Mark McReynolds, 1/11/89  

SELECT D  && schedule.dbf  

GO TOP  

SFORE 0 TO curline,page,smatches  

area=' '    && AREA Code  

stnd=' '    && STND Code  

SET EXACT OFF  

DO WHILE .NOT. EOF()  

  IF SUBSTR(d->act,5,2)='PR'  

    SELECT B  

    GO TOP  

    SEEK SUBSTR(d->act,1,9) && Substring must stay at Act,1,9 to get all matches  

    IF .NOT. FOUND()  

      IF curline>maxlinesa  

        DO header1  

      ENDIF  

      IF area <> SUBSTR(d->act,1,1)  

        area=SUBSTR(d->act,1,1)  

        DO header1  

      ENDIF  

      IF stnd <> SUBSTR(d->act,5,2)  

        stnd=SUBSTR(d->act,5,2)  

        ?' *****'+stnd+' : '+TRIM(stnd&stnd)+' *****'  

        curline=curline+1  

      ENDIF  

      ? SPACE(7)+d->act+' '+d->title+' '+DTOC(d->es)+' '+DTOC(d->ef)+' '+DTOC(d->ls);  

      +' '+DTOC(d->lf)+' '+STR(d->tf,7)+' '+STR(d->od,7)+STR(d->rd,6)+;  

      STR(d->pct,6)  

?  

      curline=curline+2  

      smatches=smatches+1  

      IF curline>maxlinesa  

        DO header1  

      ENDIF  

      area=SUBSTR(d->act,1,1)
    ENDIF

```

```
    SELECT D
ENDIF
IF EOF()
    EXIT
ENDIF
SKIP
ENDDO
? eop
?
? &footer_on
? SPACE(50)+'*** END OF REPORT B ***'
? SPACE(25)+'*** THERE WERE '+LTRIM(STR(smatches))+ ' SCHEDULE ACTIVITY ID NUMBERS WHICH MATCHED THE
SEARCH CONDITIONS. ***'
? &footer_off
? eop
RETURN
*: EOF: REPB.PRG
^Z
```

```

*:*****  

*: Program: REPC.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 23:55  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPC.PRG ~ Report C, TASK 1: PSSR/Schedule Date Comparison 1  

* Mark McReynolds, 1/11/89  

SELECT D  && schedule.dbf  

GO TOP  

STORE 0 TO curline,page,smatches,pmatches  

area=' '    && AREA Code  

stnd=' '    && STND Code  

header2=.n.  

SET EXACT OFF  

DO WHILE .NOT. EOF()  

  SELECT B  

  GO TOP  

  SEEK SUBSTR(d->act,1,9)  && Substring must stay at Act,1,9 to get all matches  

  IF FOUND()  

    DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)  

      IF CTOD(arrivesite)>d->es.AND.d->tf<10  

        IF curline>maxlinesa  

          DO header1  

        ENDIF  

        IF area <> SUBSTR(d->act,1,1)  

          area=SUBSTR(d->act,1,1)  

          DO header1  

        ENDIF  

        IF stnd <> SUBSTR(d->act,5,2)  

          stnd=SUBSTR(d->act,5,2)  

          ?'*****'+stnd+' : '+TRIM(stndv&stnd)+'*****'  

          curline=curline+1  

        ENDIF  

        ? SPACE(7)+d->act+' '+d->title+' '+DTOC(d->es)+' '+DTOC(d->ef)+' '+DTOC(d->ls);  

        +' '+DTOC(d->lf)+' '+STR(d->tf,7)+' '+STR(d->od,7)+STR(d->rd,6)+;  

        STR(d->pct,6)  

        smatches=smatches+1  

        curline=curline+1  

        IF curline>maxlinesa  

          DO header1  

        ENDIF  

        area=SUBSTR(d->act,1,1)

```

```

IF header2
? SPACE(31) +'PSSR
    ARRIVE
?? ' SCHEDULE EARLY START'
? SPACE(31) +'ACTIVITY ID      DESCRIPTION (from PSSR files)
    ON SITE      REQUIRED
?? ' - PSSR ARRIVE ON SITE'
? SPACE(31) +----- VENDOR
----- NAME
----- -----
?? ' ----- --
?
curline=curline+4
header2=.F.
ENDIF
start=RECNNO()
N=1
big=CTOD(arrivesite)
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
    IF CTOD(arrivesite)>big
        big=CTOD(arrivesite)
    ENDIF
    N=N+1
    SKIP
ENDDO
GOTO start
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
? SPACE(31)+activ_id+' '+descript+' '+vendorname+' '+arrivesite+
    '+required'
    ?
    ?? CTOD(arrivesite)-d->es
    IF N>1.AND.(CTOD(arrivesite)=big)
        ??' <-' ENDIF
        curline=curline+1
        pmatches=pmatches+1
        IF curline>maxlinesa
            DO header1
        ENDIF
        SKIP
    ENDDO
    ?
    curline=curline+1
ENDIF
SKIP
ENDDO
ENDIF
SELECT D
IF EOF()
    EXIT
ENDIF
SKIP
ENDDO
? eop

```

```
?  
? &footer_on  
? SPACE(50)+'*** END OF REPORT C ***'  
? ' *** THERE WERE '+LTRIM(STR(smatches))  
?? ' SCHEDULE ACTIVITY ID NUMBERS, WHICH MATCHED '+LTRIM(STR(pmatches))  
?  
? ' PSSR ACTIVITY ID NUMBERS, INCLUDING PSSR MULTIPLES. ***'  
? &footer_off  
? eop  
RETURN  
*: EOF: REPC.PRG  
^Z
```

```

*:*****  

*: Program: REPD.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      23:56  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPD.PRG - Report D, TASK 1: PSSR/Schedule Date Comparison 2  

* Mark McReynolds, 1/11/89  

SELECT D   && schedule.dbf  

GO TOP  

STORE 0 TO curline,page,smatches,pmatches  

area=' '    && AREA Code  

stnd=' '    && STND Code  

header2=.n.  

SET EXACT OFF  

DO WHILE .NOT. EOF()  

  SELECT B  

  GO TOP  

  SEEK SUBSTR(d->act,1,9)  && Substring must stay at Act,1,9 to get all matches  

  IF FOUND()  

    DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)  

      IF CTOD(arrivesite) > (d->es+(d->tf/5)) .AND. d->tf>10  

        IF curline>maxlinesa  

          DO header1  

        ENDIF  

        IF area <> SUBSTR(d->act,1,1)  

          area=SUBSTR(d->act,1,1)  

          DO header1  

        ENDIF  

        IF stnd <> SUBSTR(d->act,5,2)  

          stnd=SUBSTR(d->act,5,2)  

          ? ' *****' +stnd+ ' : '+TRIM(stndv&stnd)+ ' *****'  

          curline=curline+1  

        ENDIF  

        ? SPACE(7)+d->act+ ' '+d->title+ ' '+DTOC(d->es)+ ' '+DTOC(d->ef)+ ' '+DTOC(d->ls);  

        +' '+DTOC(d->lf)+ ' '+STR(d->tf,7)+ ' '+STR(d->od,7)+STR(d->rd,6)+;  

        STR(d->pct,6)  

        smatches=smatches+1  

        curline=curline+1  

        IF curline>maxlinesa  

          DO header1  

        ENDIF  

        area=SUBSTR(d->act,1,1)

```

```

IF header2
? SPACE(31)+'PSSR
    ARRIVE          SCHEDULE EARLY START'
? SPACE(31)+'ACTIVITY ID      DESCRIPTION (from PSSR files)
    ON SITE        REQUIRED      - PSSR ARRIVE ON SITE'
? SPACE(31)+'-----'          -----
-----          -----          -----
?           ?
curline=curline+4
header2=.F.

ENDIF
start=RECNO()
N=1
big=CTOD(arrivesite)
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
    IF CTOD(arrivesite)>big
        big=CTOD(arrivesite)
    ENDIF
    N=N+1
    SKIP
ENDDO
GOTO start
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
    ? SPACE(31)+activ_id+'      '+descript+'      '+vendorname+'      '+arrivesite+
    '      '+required+'      '
    ?? CTOD(arrivesite)-d->es
    IF N>1.AND. (CTOD(arrivesite)=big)
        ?? ' <-
    ENDIF
    curline=curline+1
    pmatches=pmatches+1
    IF curline>maxlinesa
        DO header1
    ENDIF
    SKIP
ENDDO
?
curline=curline+1
ENDIF
SKIP
ENDDO
ENDIF
SELECT D
IF EOF()
    EXIT
ENDIF
SKIP
ENDDO
? eop
?
? &footer_on
? SPACE(50)+'*** END OF REPORT D ***'

```

```
? ' *** THERE WERE '+LTRIM(STR(smatches))
?? ' SCEDULE ACTIVITY ID NUMBERS, WHICH MATCHED '+LTRIM(STR(pmatches))
?? ' PSSR ACTIVITY ID NUMBERS, INCLUDING PSSR MULTIPLES. ***'
? &footer_off
? eop
RETURN
*: EOF: REPD.PRG
^Z
```

```

*:*****  

*: Program: REPE.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      0:00  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPE.PRG - Report E, TASK 1: PSSR/Schedule/Submittal Comparison 1  

* Mark McReynolds, 1/11/89  

* SET STAT ON  

SELECT D  && schedule.dbf  

GO TOP  

STORE 0 TO curline,page,smatches,pmatches  

area=' '    && AREA Code  

stnd=' '    && STND Code  

header2=.n.  

SET EXACT OFF  

DO WHILE .NOT. EOF()  

  SELECT B  

  GO TOP  

  SEEK SUBSTR(d->act,1,9) && Substring must stay at Act,1,9 to get all matches  

  IF FOUND()  

    DO WHILE SUBSTR(activ_id,1,9)=SUBSTR(d->act,1,9)  

      SELECT E  

      GO TOP  

      SEEK SUBSTR(d->act,9,1)  

      IF FOUND()  

        SELECT B  

        IF CTOD(ce_submit) <> e->sch_subm .AND. CTOD(ce_subm_ap) <> e->sch_appr ;  

          .AND. d->tf< refloat  

        IF curline>maxlinesa  

          DO header1  

        ENDIF  

        IF area <> SUBSTR(d->act,1,1)  

          area=SUBSTR(d->act,1,1)  

          DO header1  

        ENDIF  

        IF stnd <> SUBSTR(d->act,5,2)  

          stnd=SUBSTR(d->act,5,2)  

          ? ' *****' +stnd+': '+TRIM(stndv&stnd)+' *****'  

          curline=curline+1  

        ENDIF  

        ? SPACE(7)+d->act+' '+d->title+' '+DTOC(d->es)+' '+DTOC(d->ef)+' '+DTOC(d->ls);  

        +' '+DTOC(d->lf)+' '+STR(d->tf,7)+' '+STR(d->od,7)+STR(d->rd,6)+;
  ENDIF

```

```

        STR(d->pct, 6)
        smatches=smatches+1
        curline=curline+1
        IF curline>maxlinesa
          DO header1
        ENDIF
        area=SUBSTR(d->act,1,1)
        IF header2
          ? SPACE(31)+'PSSR' VENDOR
          ARRIVE SCHEDULE EARLY START'
          ? SPACE(31)+'ACTIVITY ID DESCRIPTION (from PSSR files) NAME
          ON SITE REQUIRED - PSSR ARRIVE ON SITE'
          ? SPACE(31)+'-----'
          -----
          ?
          curline=curline+4
          header2=.F.
        ENDIF
        start=RECNO()
        N=1
        big=CTOD(arrivesite)
        DO WHILE SUBSTR(d->act,1,9)=SUBSTR(active_id,1,9)
          IF CTOD(arrivesite)>big
            big=CTOD(arrivesite)
          ENDIF
          N=N+1
          SKIP
        ENDDO
        GOTO start
        DO WHILE SUBSTR(d->act,1,9)=SUBSTR(active_id,1,9)
          ? SPACE(31)+active_id+' '+descript+' '+vendornamet+' '+arrivesite+';
          ' '+required+' '
          ?? CTOD(arrivesite)-d->es
          IF N>1.AND.(CTOD(arrivesite)=big)
            ?? ' <='
          ENDIF
          curline=curline+1
          pmatches=pmatches+1
          IF curline>maxlinesa
            DO header1
          ENDIF
          SKIP
          IF EOF()
            EXIT
          ENDIF
        ENDDO
        ?
        curline=curline+1
      ENDIF
      SKIP && In E
      IF EOF()
        EXIT

```

```
ENDIF
ENDIF
SELECT B
SKIP && In B
IF EOF()
    EXIT
ENDIF
ENDDO
ENDIF
SELECT D
SKIP && In D
IF EOF()
    EXIT
ENDIF
ENDDO
? eop
?
? &footer_on
? SPACE(50)+'*** END OF REPORT E ***'
? '      *** THERE WERE '+LTRIM(STR(smatches))
?? ' SCHEDULE ACTIVITY ID NUMBERS, WHICH MATCHED '+LTRIM(STR(pmatches))
?? ' PSSR ACTIVITY ID NUMBERS, INCLUDING PSSR MULTIPLES. ***'
? &footer_off
? eop
RETURN
*: EOF: REPE.PRG
^Z
```

```
*****
*:          Program: REPF.PRG
*:
*:          System: PROJECT MANAGEMENT TOOLS
*:          Author: ASC/RCE and USA-CERL
*:          Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      0:02
*:
*:          Called by: MAINX.PRG
*:
*:          Calls: HEADER1.PRG
*:
*:          Documented 04/27/89 at 10:39           SNAP! version 3.12e
*;*****
**  REPF.PRG - Report F, TASK 1: PSSR/Schedule/Submittal Comparison 2
*  Mark McReynolds, 1/11/89
SET STATUS ON
SELECT D   && schedule.dbf
GO TOP
STORE 0 TO curline,page,smatches,pmatches
area=' '    && AREA Code
stnd=' '    && STND Code
header2=.n.
SET EXACT OFF
DO WHILE .NOT. EOF()
  SELECT B
  GO TOP
  SEEK SUBSTR(d->act,1,9) && Substring must stay at Act,1,9 to get all matches
  IF FOUND()
    DO WHILE SUBSTR(activ_id,1,9)=SUBSTR(d->act,1,9)
      SELECT E
      GO TOP
      SEEK SUBSTR(d->act,9,1)
      IF FOUND()
        SELECT B
        differ=ABS(CTOD(ce_submit) - d->es)
        IF (differ/d->tf) > (rfpercent/100)
          IF curline>maxlinesa
            DO header1
          ENDIF
          IF area <> SUBSTR(d->act,1,1)
            area=SUBSTR(d->act,1,1)
            DO header1
          ENDIF
          IF stnd <> SUBSTR(d->act,5,2)
            stnd=SUBSTR(d->act,5,2)
            ? ' ****' +stnd+ ' : ' +TRIM(stndv&stnd)+ ' *****'
            curline=curline+1
          ENDIF
          ? SPACE(7)+d->act+ ' '+d->title+ ' '+DTOC(d->es)+ ' '+DTOC(d->ef)+ ' '+DTOC(d->ls)+ '
          +' '+DTOC(d->lf)+ ' '+STR(d->tf,7)+ ' '+STR(d->od,7)+STR(d->rd,6)+;
        ENDIF
      ENDIF
    ENDIF
  ENDIF
END
```

```

STR(d->pct,6)
smatches=smatches+1
curline=curline+1
IF curline>maxlinesa
DO header1
ENDIF
area=SUBSTR(d->act,1,1)
IF header2
? SPACE(31)+'PSSR
ARRIVE SCHEDULE EARLY START'
? SPACE(31)+'ACTIVITY ID DESCRIPTION (from PSSR files) NAME
ON SITE REQUIRED - PSSR ARRIVE ON SITE'
? SPACE(31)+'-----'
-----
?
curline=curline+4
header2=.n
ENDIF
r

      _1"
DO wha  _BST      .)  _LFR(activ_id,1,9)
IF CTOD(arrivesite,>L_y
big=CTOD(arrivesite)
ENDIF
N=N+1
SKIP
ENDDO
GOTO start
DO WHILE SUBSTR(d->act,1,9)=SUBSTR(activ_id,1,9)
? SPACE(31)+activ_id+' '+descript+' '+vendorname+' '+arrivesite+;
' '+required+' '
?? CTOD(arrivesite)-d->es
IF N>1 .AND. (CTOD(arrivesite)=big)
?? ' <-
ENDIF
curline=curline+1
pmatches=pmatches+1
IF curline>maxlinesa
DO header1
ENDIF
SKIP
IF EOF()
EXIT
ENDIF
ENDDO
?
curline=curline+1
ENDIF
SKIP  && In E
IF EOF()
EXIT

```

```
        ENDIF
        ENDIF
        SELECT B
        SKIP && In B
        IF EOF()
            EXIT
        ENDIF
    ENDDO
    ENDIF
    SELECT D
    SKIP && In D
    IF EOF()
        EXIT
    ENDIF
    ENDDO
? eop
?
? &footer_on
? SPCE(50)+'*** END OF REPORT F ***'
? '     *** THERE WERE '+LTRIM(STR(smatches))
?? ' SCHEDULED ACTIVITY ID NUMBERS, WHICH MATCHED '+LTRIM(STR(pmatches))
?? ' PSSR ACTIVITY ID NUMBERS, INCLUDING PSSR MULTIPLES. ***'
? &footer_off
? eop
RETURN
*: EOF: REPF.PRG
^Z
```

```

*:*****  

*:  

*:      Program: REPG.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      0:08  

*:  

*:      Called by: MAINX.PRG  

*:  

*:      Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

**  REPG.PRG - Report G, TASK 2: Resource Loading Analysis, Manpower Variance  

*  Mark McReynolds, 1/11/89  

SELECT D  && schedule.dbf  

SET EXACT OFF  

STORE 0 TO curline,page,matches  

stnd='ZZ'  && stnd=SUBS(Act,5,2)  && Get the first STND Code  

newpage=.y.  

N=0  && No. of records to be included in S.D. calculations  

men=0  && No. of men/day required for an Activity  

mean=0  && Mean no. of men/day required for an Activity  

mensq=0  && Men squared  

sdev=0  && Standard deviation  

DO WHILE .NOT. EOF()  && For D  

    I F      ( m h _ d a y / 8 ) > r g m n . A N D .  

    SUBSTR(act,5,2)$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC';  

    .AND. SUPSTR(act,5,2) <> '' .AND. pct <> 100  && If valid STND Code & not 100% Complete  

    IF stnd <> SUBSTR(act,5,2) .AND. stnd <> ''  && If STND changes  

        stnd=SUBSTR(act,5,2)  

        start=RECNO()          && Pre-calculate S.D.  

        STORE 0 TO N, mean, men, mensq  

        DO WHILE SUBSTR(act,5,2)=stnd .AND. .NOT. EOF()  

            IF (mh_day/8)>rgmn  

                men=men+(mh_day/8)  

                mensq=mensq+((mh_day/8)**2)  

                N=N+1  

            ENDIF  

            SKIP  

        ENDDO  

        mean=men/N  

        IF (N > 1)  

            sdev=SQRT((mensq-((men**2)/N))/(N-1))  && Calculate S.D.  

        ELSE  

            sdev=0.0  

        ENDIF  

        GO start          && Ready to print records  

        newpage=.y.  

    ENDIF  

    IF newpage  

        ? eop

```



```
? SPACE(20)+'*** THERE WERE '+LTRIM(STR(matches))+ ' ACTIVITY ID NUMBERS WHICH MATCHED THESE SEARCH  
CRITERIA. ***'  
? &footer_off  
? eop  
RETURN  
*: EOF: REPG.PRG  
^Z
```

```

*:*****  

*: Program: REPH.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      1:38  

*: Called by: MAINX.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPH.PRG - Report H, TASK 2: Resource Loading Analysis, Funding Variance  

* Mark McReynolds, 1/11/89  

SELECT D   && schedule.dbf  

SET EXACT OFF  

STORE 0 TO curline,page,matches  

stnd='ZZ'  && stnd=SUBS(act,5,2)  && Get the first STND Code  

newpage=.y.  

N=0  && No. of records to be included in S.D. calculations  

dol=0  && No. of dollars/day required for an Activity  

mean=0  && Mean no. of dollars/day required for an Activity  

dolsq=0  && Dollars squared  

sdev=0  && Standard deviation  

DO WHILE .NOT. EOF()  && For D
    I F          d o l _ d a y > r h d l . A N D .
    SUBSTR(act,5,2)$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC';
        .AND. SUBSTR(act,5,2) <> '' .AND. pct <> 100  && If valid STND Code & not 100% Complete
    IF stnd <> SUBSTR(act,5,2) .AND. stnd <> ''  && If STND changes
        stnd=SUBSTR(act,5,2)
        start=RECNO()  && Pre-calculate S.D.
        STORE 0 TO N, mean, dol, dolsq
        DO WHILE SUBSTR(act,5,2)=stnd .AND. .NOT. EOF()
            IF dol_day>rhdl
                dol=dol+dol_day
                dolsq=dolsq+(dol_day**2)
                N=N+1
            ENDIF
            SKIP
        ENDDO
        mean=dol/N
        IF (N > 1)
            sdev=SQRT((dolsq-((dol**2)/N))/(N-1))  && Calculate S.D.
        ELSE
            sdev=0.0
        ENDIF
        GO start      && Ready to print records
        newpage=.y.
    ENDIF
    IF newpage
        ? eop

```



```
CRITERIA. ***  
? &footer_off  
? eop  
RETURN  
*: EOF: REPH.PRG  
^Z
```

```

*:*****  

*: Program: REPI.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      1:39  

*: Called by: MAINX.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e  

*:*****  

** REPI.PRG - Report I, TASK 2: Resource Loading Analysis, Time Variance  

* Mark McReynolds, 1/11/89  

SELECT D  && schedule.dbf  

SET EXACT OFF  

STORE 0 TO curline,page,matches  

stnd='ZZ'  && stnd=SUBS(Act,5,2)  && Get the first STND Code  

newpage=.T.  

N=0      && No. of records to be included in S.D. calculations  

DAY=0    && No. of days required for an Activity  

mean=0   && Mean no. of days required for an Activity  

daysq=0  && Days squared  

sdev=0   && Standard deviation  

DO WHILE .NOT. EOF()  && For D  

  IF (od)>ridy .AND. SUBSTR(act,5,2)$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC'  

    .AND. SUBSTR(act,5,2) <> '' .AND. pct <> 100  && If valid STND Code & not 100% Complete  

    IF stnd <> SUBSTR(act,5,2) .AND. stnd <> ''  && If STND changes  

      stnd=SU  I'R(act,5,2)  

      start=RECNO()          && Pre-calculate S.D.  

      STORE 0 TO N, mean, DAY, daysq  

      DO WHILE SUBSTR(act,5,2)=stnd .AND. .NOT. EOF()  

        IF od > ridy .AND. pct <> 100  

          DAY=DAY+od  

          daysq=daysq+(od**2)  

          N=N+1  

        ENDIF  

        SKIP  

      ENDDO  

      mean=DAY/N  

      IF (N > 1)  

        sdev=SQRT((daysq-(DAY**2)/N)/(N-1))  && Calculate S.D.  

      ELSE  

        sdev=0.0  

      ENDIF  

      GO start          && Ready to print records  

      newpage= y  

    ENDIF  

    IF newpage  

      ? eop  

      curline=0

```



```
CRITERIA. ***'  
? &footer_off  
? eop  
RETURN  
*: EOF: REPI.PRG  
^Z
```

```

*:*****
*:
*:      Program: REPJ.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      6:20
*:
*:      Called by: MAINX.PRG
*:
*:      Calls: HEADER3.PRG
*:            : HEADER4.PRG
*:
*:      Indexes: SCHD_SEQ.NDX
*:            : SCHD_SRT.NDX
*:
*:      Documented 04/27/89 at 10:39           SNAP! version 3.12e
*:*****
** REPJ.PRG - TASK 2, Report J: Logical Analysis Module (checks sequence of
* construction processes) Mark McReynolds 2/22/89
page=0
SELECT D  && schedule.dbf
GO TOP
SET DATE ANSI
IF rjcode='1'  && STND Code - No need to REPlace Esa Code
  INDEX ON SUBSTR(act,2,1)+SUBSTR(act,5,2)+DTOC(es) TO schd_seq
ENDIF
IF rjcode='2' .OR. rjcode='3' .OR. rjcode='4'  && SORT Code - Need to REPlace Esa Code
  * For now, use SORT Code for all these choices
  sortstring='CE,SS,CC,CF,PR,CW,PB,PD,PF,PG,PI,PS,PW,AD,FA,EF,EW,EE,MT,FF,FP'
DO WHILE .NOT. EOF()
  DO CASE
    CASE SORT='CE'
      REPL esa WITH 'A'
    CASE SORT='SS'
      REPL esa WITH 'B'
    CASE SORT='CC'
      REPL esa WITH 'C'
    CASE SCRT='CC'
      REPL esa WITH 'D'
    CASE SORT='CF'
      REPL esa WITH 'E'
    CASE SORT='PR'
      REPL esa WITH 'F'
    CASE SORT='CW'
      REPL esa WITH 'G'
    CASE SORT='PB'
      REPL esa WITH 'H'
    CASE SORT='PD'
      REPL esa WITH 'I'
    CASE SORT='PF'

```

```

REPL esa WITH 'J'
CASE SORT='PG'
    REPL esa WITH 'K'
CASE SORT='PI'
    REPL esa WITH 'L'
CASE SORT='PS'
    REPL esa WITH 'M'
CASE SORT='PW'
    REPL esa WITH 'N'
CASE SORT='AD'
    REPL esa WITH 'O'
CASE SORT='FA'
    REPL esa WITH 'P'
CASE SORT='EF'
    REPL esa WITH 'Q'
CASE SORT='EW'
    REPL esa WITH 'R'
CASE SORT='EE'
    REPL esa WITH 'S'
CASE SORT='MT'
    REPL esa WITH 'T'
CASE SORT='FF'
    REPL esa WITH 'U'
CASE SORT='FP'
    REPL esa WITH 'V'
OTHERWISE
    REPL esa WITH ''
ENDCASE
SKIP
ENDDO
INDEX ON SUBSTR(act,2,1)+DTOC(es)+esa TO schd_seq
GO TOP
SET DATE AMERICAN
ENDIF
? eop
page=page+1
?? &intro_on
? SPACE(17) +'PROJECT MANAGEMENT TOOLS: REPORT '+reportch
?? &intro_off
?
?
?
?[
        The SORT Codes choice assumes the following sequence of construction Activity
progress:)
?[
        START --> CE : Earthwork; SS : Structural Steel; CC : Concrete; CF :
Concrete Floors; RV : Rough-In;]
?[
        WV : Dampproofing; PB : Piping; PD : Subdrains; PG : Gas Piping; PI :
Pipe Insulation; PS : Sewer]
?[
        Piping; PW : Water Piping; AD : Drywall; FA : Finish Architectural; EF :
Electrical Fixtures;]
?[
        EW : Wire & Terminals; EE : Electrical Equipment; MT : Mechanical
Equipment; FF : Carpet & Linoleum;]

```

```

?[
    PV : Paint & Wallpaper --> FINISH]
?
?
?[
    The STND Codes choice assumes the following sequence of construction Activity
progress:]
?[
    START --> PR : Procurement; SC : Subcontract Award; DR : Drawings; PM
: Prep Meetings;]
?[
    02 : Site Work; 01 : General; 03 : Concrete; 05 : Metal; 04 : Masonry;
06 : Wood and Plastics;]
?[
    07 : Moisture-Thermal Control; 08 : Doors, Windows and Glass; 09 : Finishes;
10 : Specialties;]
?[
    12 : Furnishings; 13 : Special Construction; 14 : Conveying Systems; 15
: Mechanical;]
?[
    16 : Electrical --> FINISH]

area=SUBSTR(act,1,1)  && First, find the first "AREA" code in the Schedule
IF area=' '
    ? header_on
    ? '      THERE ARE NO ACTIVITIES FOUND IN SCHEDULE FILE '+&filsched
    ? header_off
ENDIF
STORE 0 TO curline, page, matches
SET EXACT OFF

?eop
F='U'      && Current floor of building
DO WHILE .NOT. EOF() .AND. VAL(F) < 5
    F=LTRIM(F)
    SEEK F  && Find each successive floor
    IF FOUND()
        DO header3
            startef=RECNO()
            LAST=' '
            DO WHILE SUBSTR(act,2,1)=F
                IF esa <> ''
                    LAST=esa
                    lastsort=SORT
                ENDIF
                SKIP
                IF esa <> '' .AND. LAST > esa
                    y='sr'+RTRIM(SORT)
                    z='sr'+RTRIM(SUBSTR(lastsort,1,2))
                    IF TYPE(z) = 'U'
                        zz = "** Bad Code **"
                        z = "zz"
                    ENDIF
                    ? '      '+act+'      '+TITLE+'      '+DTOC(es)+'      '+DTOC(ef);
                    +'      '+DTOC(ls)+'      '+DTOC(lf)+'      '+STR(tf,7)+STR(cd,7)+STR(ed,7);
                    +'      '+STR(pct,6)+'      '+RTRIM(SORT)+': '+&y  && +STR((d_xx-es),7)
                    ? SPACE(12)+'Out-of-sequence because it has an "Early Start Date" later than that';
                    +' of an Activity on this floor with SORT code '+RTRIM(lastsort)+': '+&z
                matches=matches+1
    ENDIF

```

```

        curline=curline+2
    ENDIF
    IF curline>maxlinesj
        DO header3
    ENDIF
    ENDDO
ENDIF
F=VAL(F)
F=F+1
F=STR(F)
ENDDO
? eop

* Roofing/Drywall Summary
matches2=0
curline=0
GO TOP
SEEK '4'   && Look for earliest roof work date
latedate=es
DO WHILE SUBSTR(act,2,1)='4'
    SKIP
    IF es>latedate
        latedate=es
    ENDIF
ENDDC
GO TOP
INDEX ON SORT TO schd_srt
SEEK 'AD'
doheader4=.T.
DO WHILE SORT='AD' .AND. .NOT. EOF()
    IF es<latedate
        IF doheader4
            DO header4
            doheader4=.F.
        ENDIF
        y='sr'+RTRIM(SORT)
        ? '      '+act+'      '+TITLE+' '+esat' '+DTOC(es)+' '+DTOC(ef);
        +' '+DTOC(ls)+' '+DTOC(lf)+' '+STR(tf,7)+STR(od,7)+STR(rd,7);
        +STR(pct,6)+' '+RTRIM(SORT)+': '+&y
        matches2=matches2+1
        curline=curline+2
        IF curline>maxlinesj
            doheader4=.T.
        ENDIF
    ENDIF
    SKIP
ENDDO
IF matches2=0
    page=page+1
    ?? &header_on
    ? SPACE(17)+'PROJECT MANAGEMENT TOOLS: REPORT '+reportch
    ?

```

```
? SPACE(64) +'PAGE #' +STR(page, 6)
?
? SPACE(30) +'NO DRYWALL ACTIVITIES HAVE "EARLY START DATES" EARLIER THAN THAT OF THE EARLIEST
ROOFING ACTIVITY.'
? &header_off
ENDIF
? eop

? &footer_on
? SPACE(45) +'*** END OF REPORT J ***'
? SPACE(30) +'***** THERE WERE '+LTRIM(STR(matches))+ ' OUT-OF-SEQUENCE ACTIVITIES. *****'
? &footer_off
RETURN
*: EOF: REPJ.PRG
^Z
```

```

*:*****
*:
*:      Program: REPK.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      2:46
*:
*:      Called by: MAINX.PRG
*:
*:      Calls: FIND1.PRG
*:
*:      Indexes: SUCC_SUC.NDX
*:                  : SUCC_ACT.NDX
*:                  : SCHD_TF.NDX
*:                  : SCHD_ACT.NDX
*:
*:      Documented 04/27/89 at 10:39           SNAP! version 3.12e
*:*****
```

\* PROGRAM : REPK.PRG  
\* PROGRAMMER : Martin Hanley, Bill Nelson, John Elston  
\* PURPOSE : To find the paths through the activities

```

SET TALK OFF
SET SAFETY OFF
SET DATE ANSI
SET CENTURY ON
SET EXACT OFF
IF output $ "BP"
  SET PRINT OFF
ENDIF
IF output $ "ABD"
  SET ALTERNATE OFF
ENDIF

STORE 0 TO page
STORE .T. TO newpage
STORE 0 TO curline

@ 10,10 SAY "Sorting Input Data"
@ 24,04 SAY "REPORT K: Activities to Crash"

STORE "" TO tpath,path2,path3,path4
STORE 0 TO counter,counter1

SELECT B
INDEX ON suc TO succ_suc
INDEX ON act TO succ_act

SELECT D
INDEX ON efa+STR((200000+tf),6)+DTOC(ef) TO schd_tf

```

```

INDEX ON act TO schd_act

SELECT D
SET INDEX TO schd_tf
GO TOP

@ 10,10 SAY "Searched      of " + LTRIM(STR(RECCOUNT(),6))
DO WHILE counter < 4 .AND. .NOT. EOF()
  STORE act TO cur
  *
  * DETERMINE PATH WITH LEAST FLOAT FROM THIS POINT
  *
  SET INDEX TO schd_act
  SEEK cur
  STORE LTRIM(STR(RECNO())) TO tpath

  SELECT B
  SET INDEX TO succ_act
  SEEK cur
  DO WHILE .NOT. EOF()
    STORE suc TO tsuc
    SELECT D
    SEEK tsuc
    STORE tsuc TO msuc
    STORE tf TO mtf
    STORE RECNO() TO mrec
    SELECT B
    SKIP 1
    DO WHILE act = cur .AND. .NOT. EOF()
      STORE suc TO tsuc
      SELECT D
      SEEK tsuc
      IF .NOT. EOF()
        IF mtf > tf
          STORE tsuc TO msuc
          STORE tf TO mtf
          STORE RECNO() TO mrec
        ENDIF
      ELSE
        EXIT
      ENDIF
      SELECT B
      SKIP 1
    ENDDO
    STORE tpath + "." + LTRIM(STR(mrec)) TO tpath
    SEEK msuc
  ENDDO
  *
  * SAVE FOUND PATH & FLOAT IN APPROPRIATE VARIABLES
  *
  STORE counter + 1 TO counter
  STORE LTRIM(STR(counter)) TO tempstr

```

```

STORE tpath TO path&tempstr
*
* SKIP ACTIVITIES WHICH OCCUR IN PATHS ALREADY FOUND &
* SET RECORD POINTER TO NEXT HIGHEST FLOAT RECORD TO START PATH SEARCH
*
SELECT D
SET INDEX TO schd_tf
GOTO TOP
STORE counter1 + 1 TO counter1
SKIP counter1
STORE "." + LTRIM(STR(RECNO())) + "." TO tempstr
STORE "." + path1 + "." TO tpath1
STORE "." + path2 + "." TO tpath2
STORE "." + path3 + "." TO tpath3
DO WHILE (AT(tempstr,tpath1) > 0) .OR. (AT(tempstr,tpath2) > 0) .OR. ;
      (AT(tempstr,tpath3) > 0)
      STORE counter1 + 1 TO counter1
      SKIP 1
      STORE "." + LTRIM(STR(RECNO())) + "." TO tempstr
ENDDO
*
* REINITIALIZE PATH VARIABLE
*
STORE "" TO tpath
@ 10,19 SAY STR(counter1,6)
ENDDO    && END OF MAIN PROCESSING LOOP
*
* OUTPUT ANY PATHS FOUND
*
IF output $ "BP"
  SET PRINT ON
ENDIF
IF output $ "ABD"
  SET ALTERNATE ON
ENDIF
SET CENTURY OFF
SET DATE AMERICAN
STORE 1 TO prthead
SELECT D
SET INDEX TO
DO find1 WITH path1,1
DO find1 WITH path2,2
DO find1 WITH path3,3
DO find1 WITH path4,4

RETURN

*: EOF: REPK.PRG
^Z

```



```

newpage=.F.
ENDIF

IF tf > 0
  IF tf>ufloat .AND. ((f->tf - (upercnt * f->tf)) > tf)
    ? act+' '+TITLE+
    ?? DTOC(es)+ ' '+DTOC(ef)+' '+DTOC(ls)+' '+DTOC(lf)+' '+DTOC(as)+' '+DTOC(af)+'
    ?? STR(tf,6)+' '+STR(od,6)+' '+STR(rd,6)+' '+STR(f->od,6)+' '+STR(pct,6,2)
    curline=curline+1
    matches=matches+1
  ENDIF
ENDIF
SKIP
IF curline>maxlines1
  newpage=.T.
  EXIT
ENDIF

ENDIF
SELECT F
SKIP
IF EOF()
  EXIT
ENDIF
ENDDO
ENDIF

? eop
?
?
? footer_on
? SPACE(50)+'*** END OF REPORT L ***'
? SPACE(20)+'*** THERE WERE '+LTRIM(STR(matches))+ ACTIVITY ID NUMBERS WHICH MATCHED THESE SEARCH
CRITERIA. ***
? footer_off
? eop
RETURN
`: EOF: REPL.PRG
`z

```

```

*:*****  

*: Program: REPM.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89 1:43  

*: Called by: MAINX.PRG  

*: Calls: HEADER1.PRG  

*: Documented 04/27/89 at 10:39           SNAP! version 3.12e
*:*****  

** REPM.PRG - Do Report M  

STORE 0 TO curline,page  

STORE 0 TO schdtot, pssrtot, submtot, schdmat, pssrmat, submmat  

STORE 0 TO schdprrtot, pssrprrtot, submprtmat, schdprmat, pssrprmat, submprmat  

STORE 0 TO rduration, oduration  

STORE 0 TO badstndcod,badareacod,badsuarcod,badtypecod,badsortcod,badrespcod,;  

    badlvsccod,badsubscod  

STORE 0 TO negfloat, critpattot, critpates, critpatpct, asilate, pctover0, rem_ordinalat  

area=' '    && AREA Code  

stnd=' '    && STND Code  

header2=F.  

path1 = "."+path1+"."
path2 = "."+path2+"."
path3 = "."+path3+"."
path4 = "."+path4+"."
SET EXACT OFF
SELECT B
GO TOP
pssrtot=RECCOUNT()
DO WHILE .NOT. EOF()
    IF SUBSTR(activ_id,5,2)='PR'  && If this is a PRocurement Activity
        pssrprrtot=pssrprrtot+1
    ENDIF
    SKIP
ENDDO
GO TOP
SELECT E
GO TOP
submtot=RECCOUNT()
DO WHILE .NOT. EOF()
    IF SUBSTR(nas,5,2)='PR'  && If this is a PRocurement Activity
        submprtmat=submprtmat+1
    ENDIF
    SKIP
ENDDO
GO TOP
SELECT D  && schedule.dbf

```

```

GO TOP
schdtot=RECCOUNT()
DO WHILE .NOT. EOF()
  IF SUBSTR(act,5,2)='PR'  && If this is a PRocurement Activity
    schdprrtot=schdprrtot+1
  ELSE
    IF rd>20
      rduration=rduration+1
    ENDIF
    IF od>20
      oduration=oduration+1
    ENDIF
  ENDIF
  IF .NOT. SUBSTR(act,5,2)$stndcodes  && If not a valid "STND" ID Code
    badstndcod=badstndcod+1
  ENDIF
  IF .NOT. SUBSTR(act,1,1)$areacodes  && If not a valid "AREA" ID Code
    badareacod=badareacod+1
  ENDIF
  IF .NOT. SUBSTR(act,2,1)$suarcodes  && If not a valid "SUBArea" ID Code
    badsuarcod=badsuarcod+1
  ENDIF
  IF .NOT. SUBSTR(act,3,2)$typecodes  && If not a valid "TYPE of work" ID Code
    badtypecod=badtypecod+1
  ENDIF
  IF .NOT. RTRIM(SORT)$respcodes  && If not a valid "SORT" Code.
    badsortcod=badsortcod+1
  ENDIF
  IF .NOT. RTRIM(resp)$respcodes  && If not a valid "RESP" Code.
    badrespcod=badrespcod+1
  ENDIF
  IF .NOT. RTRIM(lvls)$lvlscodes  && If not a valid "LVLS" Code.
    badlvscod=badlvscod+1  && lvlscodes='D,DC,DP,DZ,JO,M,OUT,S,Z'
  ENDIF
  IF SUBS <> SPACE(5)
    IF .NOT. RTRIM(SUBS) $ subscodes1 .AND. .NOT. RTRIM(SUBS) $ subscodes2
      * If not a valid "SUBS" Code.
      badsubscod=badsubscod+1
    ENDIF
  ENDIF
  IF tf<0
    negfloat=negfloat+1
  ENDIF
  mrecno = "." + LTRIM(STR(RECNO())) + "."
  IF mrecno $ path1 .OR. mrecno $ path2 .OR. mrecno $ path3 .OR. mrecno $ path4
    critpattot=critpattot+1
  IF es>=as  && If Activity started on or before the Early Start Date
    critpates=critpates+1
  ENDIF
  IF pct>=((od/rd)*100)
    critpatpct=critpatpct+1
  ENDIF

```

```

ENDIF
IF as<(datadate-7) .AND. pct<100
  asdate=asdate+1
ENDIF
IF pct>0 (.OR. rd>0 .AND. as=0)
  pctover0=pctover0+1
ENDIF
IF rd>od    && If Remaining Duration > Original Duration
  rem_oxidat=rem_oxidat+1
ENDIF
SKIP
ENDDO
GO TOP
DO WHILE .NOT. EOF()
  SELECT B
  SEEK SUBSTR(d->act,1,9)  && Substring must stay at Activ_id,1,9 to get all matches
  IF FOUND()
    pssrmat=pssrmat+1
    IF curline>maxlinesa.AND.mch$'DB'
      DO header1
    ENDIF
    IF area <> SUBSTR(d->act,1,1) .AND. area <> '4' .AND. mch $ 'DB'
      area=SUBSTR(d->act,1,1)
      DO header1
    ENDIF
    IF stnd <> SUBSTR(d->act,5,2) .AND. mch $ 'DB'
      stnd=SUBSTR(d->act,5,2)
      ?' *****' +stnd+': '+TRIM(stndv&stnd)+;
      ' *****'
      curline=curline+1
    ENDIF
    IF mch$'DB'
      ? SPACE(7)+d->act+' '+d->title+' '+DTOC(d->es)+' '+DTOC(d->ef)+' '+DTOC(d->ls);
      +' '+DTOC(d->lf)+STR(d->tf,8)+STR(d->od,8)+STR(d->rd,6)+STR(d->pct,6)
      curline=curline+2
      ?
      IF curline>maxlinesa
        DO header1
      ENDIF
      area=SUBSTR(d->act,1,1)
    ENDIF
    IF SUBSTR(d->act,5,2)='PR'  && If this is a PRocurement Activity
      schdprmat=schdprmat+1
    ENDIF
    IF SUBSTR(activ_id,5,2)='PR'  && If this is a PRocurement Activity
      psrprmat=psrprmat+1
    ENDIF
  ENDIF
  SELECT E
  GO TOE
  IF EOF()
    EXIT

```

```

ENDIF
mnas=SUBSTR(nas,1,9)
SEEK SUBSTR(d->act,1,9)  && Substring must stay at Activ_id,1,9 to get all matches
IF FOUND()
  submmat=submmat+1
  IF SUBSTR(e->nas,5,2)='PR'  && If this is a PRocurement Activity match
    submprmat=submprmat+1
  ENDIF
ENDIF
SELECT D
IF EOF()
  EXIT
ENDIF
mact=SUBSTR(act,1,9)
SKIP
ENDDO
? eop
?? &header_on
? SPACE(17)+' SCHEDULING ANALYSIS SYSTEM: REPORT '+reportch
? &header_mid
? SPACE(50)+'*'+' SUMMARY FOR REPORT M '+'*'+' PAGE 1'
?
?
? SPACE(40)+' ITEM ONE, PART A: Overall Activity Correspondence'
?
? '      This Schedule file, '+filsched+', contains ';
+LTRIM(STR(schdtot))+'' unique Activity ID numbers.'
? '      The PSSR file, '+filpssr_pr+', contains ';
+LTRIM(STR(pssrtot))+'' unique Activity ID numbers.'
? '      The Submittal file, '+filsubmit+', contains ';
+LTRIM(STR(submtot))+'' unique Activity ID numbers.'
testvar=(pssrmat/pssrtot*100)
SELECT F
GOTO TOP
IF testvar>a2
  SKIP
  IF testvar>a2
    SKIP
    IF testvar>a2
      SKIP
      IF testvar>a2
        SKIP
        ENDIF
      ENDIF
    ENDIF
  ENDIF
? SPACE(10)+LTRIM(STR(pssrmat))+'' PSSR Activities, or ';
+LTRIM(STR(pssrmat/pssrtot*100,10,2))+''% of the total PSSR Activities,';
'+'' matched Activities in the Schedule file. GRADE = '+LTRIM(category)
testvar=(submmat/submtot*100)
GOTO TOP
IF testvar>a2

```

```

SKIP
IF testvar>a2
  SKIP
  IF testvar>a2
    SKIP
    IF testvar>a2
      SKIP
      ENDIF
    ENDIF
  ENDIF
ENDIF
? SPACE(10)+LTRIM(STR(submmat))+ ' Submittal Activities, or ';
+LTRIM(STR(submmat/submtot*100,10,2))+ '% of the total Submittal Activities,';
+' matched Activities in the Schedule file. GRADE = '+LTRIM(category)
? SPACE(15)+ '(According to the grading scale input earlier, this assumes that the"';
+RTRIM(category)+'" range is from '+LTRIM(STR(a1))+ '% to '+LTRIM(STR(a2))+ '%.)'
?
?
? SPACE(40)+ 'ITEM ONE, PART B: Procurement Activity Correspondence'
?
?' This Schedule file, '+filsched+, contains '+LTRIM(STR(schdprtot));
+' unique Activity ID numbers which indicate Procurement Activities.'
?' The PSSR file, '+filpssr_pr+, contains '+LTRIM(STR(pssrprtot));
+' unique Activity ID numbers which indicate Procurement Activities.'
?' The Submittal file, '+filsubmi+, contains '+LTRIM(STR(submprtot));
+' unique Activity ID numbers which indicate Procurement Activities.'
testvar=(pssrprmat/schdprtot*100)
GOTO TOP
IF testvar>b2
  SKIP
  IF testvar>b2
    SKIP
    IF testvar>b2
      SKIP
      IF testvar>b2
        SKIP
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
? SPACE(10)+LTRIM(STR(pssrprmat))+ ' Schedule Procurement Activities, or ';
+LTRIM(STR(pssrprmat/schdprtot*100,10,2))+ '% of the total Schedule Procurement Activities,';
? SPACE(12)+ ' matched Procurement Activities in the PSSR file. GRADE = '+LTRIM(category)
testvar=(submprtmat/schdprtot*100)

GOTO TOP

IF testvar>b2
  SKIP
  IF testvar>b2
    SKIP
    IF testvar>b2

```

```

    SKIP
    IF testvar>b2
        SKIP
    ENDIF
    ENDIF
    ENDIF
    ENDIF

? SPACE(10)+LTRIM(STR(submprmat))+ ' Schedule Procurement Activities, or '
+LTRIM(STR(submprmat/schdprt*100,10,2))+ '% of the total Schedule Procurement Activities.'
? SPACE(12)+ ' matched Procurement Activities in the Submittal file. GRADE = '+LTRIM(category)
? SPACE(15)+ '(According to the grading scale input earlier, this assumes that the ';
+RTRIM(category)+ " range is from "+LTRIM(STR(b1))+ "% to "+LTRIM(STR(b2))+ "%.)'

?
?

? SPACE(40)+'ITEM ONE, PART C: Non-Procurement Activity Durations'
?

?' This Schedule file, '+filsched+, contains ';
+LTRIM(STR(schdtot-schdprt*100,10,2))+ ' non-Procurement Activities.'
testvar=(rduration/(schdtot-schdprt)*100)

GOTO TOP

IF testvar>c2
    SKIP
    IF testvar>c2
        SKIP
        IF testvar>c2
            SKIP
            IF testvar>c2
                SKIP
                ENDIF
            ENDIF
        ENDIF
    ENDIF

? SPACE(10)+LTRIM(STR(rduration))+', or '+LTRIM(STR(rduration/(schdtot-schdprt)*100,10,2));
+'%, of Schedule non-Procurement Activities have "Remaining Durations" exceeding 20 days. GRADE
= '+LTRIM(category)
testvar=(oduration/(schdtot-schdprt)*100)

GOTO TOP

IF testvar>c2
    SKIP
    IF testvar>c2
        SKIP
        IF testvar>c2
            SKIP
            IF testvar>c2
                SKIP
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF

```

```

ENDIF
? SPACE(10)+LTRIM(STR(oduration))+', or '+LTRIM(STR(oduration/(schdtot-schdprtot)*100,10,2));
+'%, of Schedule non-Procurement Activities have "Original Durations" exceeding 20 days. GRADE
= '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the ''';
+RTRIM(category)+" range is from '+LTRIM(STR(c2))+%' to '+LTRIM(STR(c1))+%.)'
?
?
? SPACE(40)+'ITEM ONE, PART D: Correct Coding '
?

?' This Schedule file, '+filsched+', contains '+LTRIM(STR(schdtot));
+' Activity ID numbers. The following Activity Codes and Activtiy ID Codes'
?' are analyzed for their compliance with the Code Dictionary';
+' developed for the "NTF" system in Primavera.'
testvar=((schdtot-badstndcod)/schdtot*100)

GOTO TOP

IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badstndcod))+', or ';
+LTRIM(STR((schdtot-badstndcod)/schdtot*100,10,2));+%, have the correct "STND" ID Code. GRADE =
'+LTRIM(category)
testvar=((schdtot-badareacod)/schdtot*100)

GOTO TOP

IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badareacod))+', or ';

```

```
+LTRIM(STR((schdtot-badareacod)/schdtot*100,10,2))+%, have the correct "AREA" ID Code. GRADE =
'+LTRIM(category)
testvar=(schdtot-badsuarcod)/schdtot*100)
```

```
GOTO TOP
```

```
IF testvar>d2
    SKIP
    IF testvar>d2
        SKIP
        IF testvar>d2
            SKIP
            IF testvar>d2
                SKIP
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

```
? SPACE(10)+LTRIM(STR(schdtot-badsuarcod))+', or ';
+LTRIM(STR((schdtot-badsuarcod)/schdtot*100,10,2))+%, have the correct "SUBAREA" ID Code. GRADE
= '+LTRIM(category)
testvar=(schdtot-badtypecod)/schdtot*100)
```

```
GOTO TOP
```

```
IF testvar>d2
    SKIP
    IF testvar>d2
        SKIP
        IF testvar>d2
            SKIP
            IF testvar>d2
                SKIP
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

```
? SPACE(10)+LTRIM(STR(schdtot-badtypecod))+', or ';
+LTRIM(STR((schdtot-badtypecod)/schdtot*100,10,2))+%;
%, have the correct "TYPE OF WORK" ID Code. GRADE = '+LTRIM(category)
testvar=(schdtot-badsortcod)/schdtot*100)
```

```
GOTO TOP
```

```
IF testvar>d2
    SKIP
    IF testvar>d2
        SKIP
        IF testvar>d2
            SKIP
```

```

    IF testvar>d2
        SKIP
    ENDIF
    ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badsortcod))+', or ';
+LTRIM(STR((schdtot-badsortcod)/schdtot*100,10,2))+;
', have the correct "SORT" Code. GRADE = '+LTRIM(category)
testvar=((schdtot-badrespcod)/schdtot*100)

GOTO TOP

IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badrespcod))+', or ';
+LTRIM(STR((schdtot-badrespcod)/schdtot*100,10,2))+%', have the correct "RESPONSIBILITY" Code.
GRADE = '+LTRIM(category)
testvar=((schdtot-badlvlscod)/schdtot*100)

GOTO TOP

IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badlvlscod))+', or ';
+LTRIM(STR((schdtot-badlvlscod)/schdtot*100,10,2))+%', have the correct "LVLS" Code. GRADE =
'+LTRIM(category)
testvar=((schdtot-badsubscod)/schdtot*100)

GOTO TOP

```

```

IF testvar>d2
    SKIP
IF testvar>d2
    SKIP
IF testvar>d2
    SKIP
IF testvar>d2
    SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? SPACE(10)+LTRIM(STR(schdtot-badsubscod))+', or ';
+LTRIM(STR((schdtot-badsubscod)/schdtot*100,10,2))+;
'% have the correct "SUBCONTRACTORS" Code. GRADE = '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the"';
+RTRIM(category)+'" range is from '+LTRIM(STR(d1))+'% to '+LTRIM(STR(d2))+'.)'

? eop
?? &header_on
? SPACE(17)+' SCHEDULING ANALYSIS SYSTEM: REPORT '+reportch
? &header_mid
? SPACE(50)+'*** SUMMARY FOR REPORT M ***'+SPACE(45)+'PAGE 2'
?
?
? SPACE(40)+' ITEM TWO, PART A: Negative Float'
?
SELECT G
GOTO TOP

testvar=(negfloat/schdtot*100)
IF testvar<a2
    SKIP
IF te:tvar<a2
    SKIP
IF testvar<a2
    SKIP
IF testvar<a2
    SKIP
ENDIF
ENDIF
ENDIF
ENDIF

?' This Schedule file, '+filsched+', contains '+LTRIM(STR(schdtot))+ Activities, of which ';
+LTRIM(STR(negfloat))+', or '+LTRIM(STR(negfloat/schdtot*100,10,2))+% have negative float.
GRADE = '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the"';
+RTRIM(category)+'" range is from '+LTRIM(STR(a2))+'' to '+LTRIM(STR(a1))+
? SPACE(15)+' Activities with negative float.)'
?

```

```

?
? SPACE(40)+'ITEM TWO, PART B: Critical Path Analysis 1'
?
IF critpattot=0
?' IN ORDER TO GET MEANINGFUL OUTPUT FOR THIS SUB-ITEM, REPORT K:, "ACTIVITIES TO CRASH", MUST
BE RUN FIRST.'
ELSE

GOTO TOP

testvar=(critpates/critpattot*100)
IF testvar>b2
SKIP
IF testvar>b2
SKIP
IF testvar>b2
SKIP
IF testvar>b2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

?' In Schedule file '+filsched+', '+LTRIM(STR(critpates))+ of ';
'+LTRIM(STR(critpattot))+ Activities on the four most critical paths,';
+' or '+LTRIM(STR(critpates/critpattot*100,10,2))+%, started'
?' on their "Early Start Date". GRADE = '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the "";
+RTRIM(category)+" range is from '+LTRIM(STR(b1))+% to '+LTRIM(STR(b2))+%).'
ENDIF

?

?

? SPACE(40)+'ITEM TWO, PART C: Critical Path Analysis 2'
?

IF critpattot=0
?' IN ORDER TO GET MEANINGFUL OUTPUT FOR THIS SUB-ITEM, REPORT K:, "ACTIVITIES TO CRASH", MUST
BE RUN FIRST.'
ELSE

GOTO TOP
testvar=(critpatpct/critpattot*100)
IF testvar>c2
SKIP
IF testvar>c2
SKIP
IF testvar>c2
SKIP
IF testvar>c2
SKIP
ENDIF

```

```

ENDIF
ENDIF
ENDIF
? ' In Schedule file '+filsched+', '+LTRIM(STR(critpatpct))+' of ';
+LTRIM(STR(critpattot))+ Activities on the four most critical paths,';
+' or '+LTRIM(STR(critpatpct/critpattot*100,10,2))+'%,'
? ' have a "Percent Complete" value greater than or equal to the';
+' proportion of the "Original Duration" value to'
? ' that of the "Remaining Duration". GRADE = '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the"';
+RTRIM(category)+'" range is from '+LTRIM(STR(c1))+'% to '+LTRIM(STR(c2))+%).'
ENDIF

?
?

? SPACE(40)+'ITEM TWO, PART D: "Actual Start Date" Analysis'
?

GOTO TOP

testvar=(asdate/schdtot*100)
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
IF testvar>d2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? ' This Schedule file, '+filsched+, contains '+LTRIM(STR(schdtot));
+' Activities, of which '+LTRIM(STR(asdate))+', or ';
+LTRIM(STR(asdate/schdtot*100,10,2))+'% will be completed'
? ' if the "Actual Start Date" is greater than 1 week prior to the ';
+' Primavera "Data Date", and the "Percent Completed" is'
? ' less than 100. GRADE = '+LTRIM(category)
? SPACE(15)+'(According to the grading scale input earlier, this assumes that the"';
+RTRIM(category)+'" range is from '+LTRIM(STR(d1))+'% to '+LTRIM(STR(d2))+%).'
?
?

? SPACE(40)+'ITEM TWO, PART E: "Percent Complete" Greater Than 0'
?

GOTO TOP

testvar=(pctover0/schdtot*100)
IF testvar<e2
SKIP

```

```

IF testvar<e2
SKIP
IF testvar<e2
SKIP
IF testvar<e2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? ' This Schedule file, '+fileched+, contains '+LTRIM(STR(schdtot));
+' Activities, of which '+LTRIM(STR(pctover0))+', or ';
+LTRIM(STR(pctover0/schdtot*100,10,2))+% have either a "Percent Complete"
? ' greater than 0 or a "Remaining Duration" with no "Actual Start Date." GRADE =
'+LTRIM(category)
? SPACE(15)+' (According to the grading scale input earlier, this assumes that the ';
+RTRIM(category)+" range is from '+LTRIM(STR(e2))+%' to '+LTRIM(STR(e1))+%.)'
?
?
? SPACE(40)+' ITEM TWO, PART F: "Remaining Duration" Greater than "Original Duration"'
?

GOTO TOP

testvar=(rem_oridat/schdtot*100)
IF testvar>f2
SKIP
IF testvar>f2
SKIP
IF testvar>f2
SKIP
IF testvar>f2
SKIP
ENDIF
ENDIF
ENDIF
ENDIF

? ' This Schedule file, '+fileched+, contains '+LTRIM(STR(schdtot));
+' Activities, of which '+LTRIM(STR(rem_oridat))+', or ';
+LTRIM(STR(rem_oridat/schdtot*100,10,2))+% have a "Remaining Duration"
? ' greater than their "Original Duration". GRADE = '+LTRIM(category)
? SPACE(15)+' (According to the grading scale input earlier, this assumes that the ';
+RTRIM(category)+" range is from '+LTRIM(STR(f2))+%' to '+LTRIM(STR(f1))+%.)'
?
? &footer_off
? eop
RETURN
*: EOF: REPM.PRG
^Z

```

```

*:*****  

*: Program: REPO.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/20/89      14:31  

*: Called by: MAINX.PRG  

*: Calls: STND_FND.PRG  

*: Uses: NEW_TEMP.DBF  

*: : NEWSCHED.DBF  

*: : MRG_FILE.DBF  

*: : &FILSCHED1  

*: : &FILSCHED2  

*: : &FILSCHED3  

*: : &FILSCHED4  

*: : &FILSCHED5  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

CLOSE DATABASES  

CLEAR  

@ 10,10 SAY "MERGING FILES PLEASE WAIT"  

ERASE mrgndx.ndx  

SELECT G  

ERASE newsched.dbf  

USE new_temp  

COPY STRUCTURE TO newsched  

SELECT H  

USE mrg_file  

ZAP

IF LEN(TRIM(filsched1)) > 0
  STORE filsched1 TO fil_name
  APPEND FROM &filsched1
  GOTO TOP
  DO WHILE .NOT. EOF()
    REPL FILE WITH fil_name
    SKIP
  ENDDO
ENDIF

IF LEN(TRIM(filsched2)) > 0
  STORE filsched2 TO fil_name
  GOTO BOTTOM
  recnum = RECNO() + 1
  APPEND FROM &filsched2
  GO recnum

```

```

DO WHILE .NOT. EOF()
  REPL FILE WITH fil_name
  SKIP
ENDDO
ENDIF

IF LEN(TRIM(filsched3)) > 0
  STORE filsched3 TO fil_name
  GOTO BOTTOM
  recnum = RECNO() + 1
  APPEND FROM &filsched3
  GO recnum
  DO WHILE .NOT. EOF()
    REPL FILE WITH fil_name
    SKIP
  ENDDO
ENDIF

IF LEN(TRIM(filsched4)) > 0
  STORE filsched4 TO fil_name
  GOTO BOTTOM
  recnum = RECNO() + 1
  APPEND FROM &filsched4
  GO recnum
  DO WHILE .NOT. EOF()
    REPL FILE WITH fil_name
    SKIP
  ENDDO
ENDIF

IF LEN(TRIM(filsched5)) > 0
  STORE filsched5 TO fil_name
  GOTO BOTTOM
  recnum = RECNO() + 1
  APPEND FROM &filsched5
  GO recnum
  DO WHILE .NOT. EOF()
    REPL FILE WITH fil_name
    SKIP
  ENDDO
ENDIF

ERASE mrgndx.ndx

SELECT H

DO stnd_fnd

SELECT H
ZAP
RETURN

```

\*: EOF: REPO.PRG  
^Z

```

*:*****  

*:  

*:      Program: REPR.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*:      Last modified: 04/26/89      18:16  

*:  

*:      Called by: MAINX.PRG  

*:  

*:      Calls: OPENFILS      (procedure in PROCS.PRG)  

*:              : TITLE1.PRG  

*:              : PRINTACT.PRG  

*:              : CLOSEFIL      (procedure in PROCS.PRG)  

*:  

*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfiles  

*  

** FIND DELETED ACTIVITIES  

*  

** INITIALIZE AND PRINT TITLE  

*  

CLEAR  

@ 10,10 SAY "RUNNING REPORT R -- FIND DELETED ACTIVITIES"  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

*  

STORE SPACE(10) TO wskey  

STORE 0 TO Line,num,page  

STORE '1' TO rpt1  

STORE '0' TO rpt2  

DO title1  

*  

SELECT 1  

DO WHILE .NOT. EOF()  

  STORE act TO wskey  

  SELECT 2  

  FIND &wskey  

  IF EOF()  

    SELECT 1  

    num = num + 1  

    DO printact  

  ENDIF  

  SELECT 1  

  SKIP 1  

ENDDO  

Line = Line + 2  

?

```

```
? 'THE TOTAL NUMBER OF DELETED ACTIVITIES = '+ STR(num,3)
DO closefil
*DO TOTOP
RETURN
*: EOF: REPR.PRG
^Z
```

```

*:*****  

*: Program: REPR.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 18:16  

*: Called by: MAINX.PRG  

*: Calls: OPENFILS      (procedure in PROCS.PRG)  

*:        : TITLE1.PRG  

*:        : PRINTACT.PRG  

*:        : CLOSEFIL      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfils  

*  

** FIND DELETED ACTIVITIES  

*  

** INITIALIZE AND PRINT TITLE  

*  

CLEAR  

G 10,10 SAY "RUNNING REPORT R -- FIND DELETED ACTIVITIES"  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

*  

STORE SPACE(10) TO wskey  

STORE 0 TO Line,num,page  

STORE '1' TO rpt1  

STORE '0' TO rpt2  

DO title1  

*  

SELECT 1  

DO WHILE .NOT. EOF()  

  STORE act TO wskey  

  SELECT 2  

  FIND &wskey  

  IF EOF()  

    SELECT 1  

    num = num + 1  

    DO printact  

  ENDIF  

  SELECT 1  

  SKIP 1  

ENDDO  

Line = Line + 2  

?

```

```
? 'THE TOTAL NUMBER OF DELETED ACTIVITIES = '+ STR(num,3)
DO closefil
*DO TOTOP
RETURN
*: EOF: REPR.PRG
^Z
```

```

*:*****  

*: Program: REPT.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      6:48  

*: Called by: MAINX.PRG  

*: Calls: OPENFILS      (procedure in PROCS.PRG)  

*:        : TITLE1.PRG  

*:        : PRINTACT.PRG  

*:        : CLOS2YIL      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfils  

*  

** IN CURRENT SCHEDULE FIND ACTIVITIES WHICH HAVE INCONSISTENCY  

** BETWEEN PERCENT COMPLETES AND ACTUAL START/FINISH DATES  

*  

** INITIALIZE AND PRINT TITLE  

CLEAR  

@ 10,5 SAY "RUNNING REPORT T -- FIND ACTIVITIES WHICH HAVE INCONSISTENCY"  

@ 11,5 SAY "BETWEEN PERCENT COMPLETES AND ACTUAL START/FINISH DATES"  

*  

SELECT 2  

GOTO TOP  

*  

STORE 0 TO num,page,Line  

STORE '3' TO rpt1  

STORE '1' TO rpt2  

DO title1  

*  

DO WHILE .NOT. EOF()  

  IF pct = 0 .AND. DTOC(as) <> ' / / '  

    comments = 'HAS ACTUAL START DATE BUT NO PERCENT COMPLETE'  

    DO printact  

    num = num + 1  

    SKIP  

    LOOP  

  ENDIF  

*  

  IF pct > 0 .AND. DTOC(as) = ' / / '  

    comments = 'HAS NON-ZERO PCT BUT NO ACTUAL START DATE'  

    DO printact  

    num = num + 1  

    SKIP  

    LOOP  

  ENDIF

```

```
*  
IF pct = 100 .AND. DTOC(af) = ' / / '  
  comments = 'HAS 100% PCT BUT NO ACTUAL FINISH DATE'  
  DO printact  
    num = num + 1  
    SKIP  
    LOOP  
  ENDIF  
  SKIP 1  
ENDDO  
Line = Line + 2  
?  
? 'THE TOTAL NUMBER OF PCT/DATE INCONSISTENT ACTIVITIES = '+STR(num, 3)  
*DO TOTOP  
DO closefil  
RETURN  
*: EOF: REPT.PRG  
^Z
```

```

*:*****  

*:  

*:      Program: REPU.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89    18:13  

*:  

*:      Called by: MAINX.PRG  

*:  

*:      Calls: OPENFILS      (procedure in PROCS.PRG)  

*:            : TITLE1.PRG  

*:            : PRINTACT.PRG  

*:            : CLOSEFIL      (procedure in PROCS.PRG)  

*:  

*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfil$  

*  

** FIND DESCRIPTION CHANGED ACTIVITIES  

*  

** INITIALIZE AND PRINT TITLE  

*  

*  

CLEAR  

@10,10 SAY "REPORT U -- FIND DESCRIPTION CHANGED ACTIVITIES"  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

*  

STORE SPACE(10) TO wskey  

STORE 0 TO Line,num,page  

STORE '4' TO rpt1  

STORE '0' TO rpt2  

DO title1  

Line = Line + 1  

*  

SELZCT 1  

DO WHILE .NOT. EOF()  

  STORE act TO wskey  

  STORE TITLE TO wstitle  

  SELECT 2  

  FIND &wskey  

  IF .NOT. EOF()  

    *  

    ** CHANGE OF TITLE  

    *  

    IF TITLE <> wstitle  

      DO printact  

      SELECT 1

```

```
DO printact
Line = Line + 1
num = num +1
SKIP 1
LOOP
ENDIF
ENDIF
SELECT 1
SKIP 1
ENDDO
*
Line = Line + 2
?
? 'THE TOTAL NUMBER OF ACTIVITIES FOUND = ' + STR(num,3)
*DO TOTOP
DO closefil
RETURN

*: EOF: REPU.PRG
^Z
```

```

*:*****  

*: Program: REPV.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 18:14  

*: Called by: MAINX.PRG  

*: Calls: OPENFILS      (procedure in PROCS.PRG)  

*:        : TITLE1.PRG  

*:        : PRINTACT.PRG  

*:        : CLOSEFIL      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfils  

*  

** FIND ORIGINAL DURATION CHANGED ACTIVITIES  

*  

** INITIALIZE AND PRINT TITLE  

*  

CLEAR  

@ 10,10 SAY "RUNNING REPORT V -- FIND ORIGINAL DURATION CHANGED ACTIVITIES"  

  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

*  

STORE SPACE(10) TO wskey  

STORE 0 TO Line,num,page  

STORE '5' TO rpt1  

STORE '1' TO rpt2  

comments = ''  

DO title1  

Line = Line + 1  

*  

STORE 0 TO temp  

SELECT 1  

DO WHILE .NOT. EOF()  

  STORE act TO wskey  

  STORE od TO wsod  

  SELECT 2  

  FIND wskey  

  IF .NOT. EOF()  

    *  

    ** NEW OD > OLD OD AND NEAR CRITICAL PATH  

    *  

    IF od > wsod .AND. tf < 5 .AND. pct < 100

```

```
temp = (od - wsod) / od * 100
IF temp > 10
  temp2 = STR(temp)
  comments = 'INCREASED RATIO IS ' + LTRIM(temp2) + ' PERCENT'
  DO printact
  SELECT 1
  comments = ''
  DO printact
  Line = Line + 1
  num = num +1
  SKIP 1
  LOOP
ENDIF
ENDIF
ENDIF
SELECT 1
SKIP 1
ENDDO
Line = Line + 2
?
? 'THE TOTAL NUMBER OF ACTIVITIES FOUND = ' + STR(num,3)
*DO TOTOP
DO closefil
RETURN
*: EOF: REPV.PRG
^Z
```

```

*:*****  

*: Program: REPW.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 18:15  

*: Called by: MAINX.PRG  

*: Calls: OPENFILS      (procedure in PROCS.PRG)  

*:        : TITLE1.PRG  

*:        : PRINTACT.PRG  

*:        : CLOSEFIL      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfiles  

*  

** PROGRESS CHANGED ACTIVITIES  

*  

** INITIALIZE AND PRINT TITLE  

*  

CLEAR  

@ 10,10 SAY "RUNNING REPORT W -- PROGRESS CHANGED ACTIVITIES"  

  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

*  

STORE SPACE(10) TO wskey  

STORE 0 TO Line,num,page  

STORE '6' TO rpt1  

STORE '1' TO rpt2  

DO title1  

Line = Line + 1  

*  

STORE 0 TO temp  

SELECT 1  

DO WHILE .NOT. EOF()  

  STORE act TO wskey  

  STORE as TO wsas  

  STORE af TO wsaf  

  STORE pct TO wspct  

  SELECT 2  

  FIND &wskey  

  IF .NOT. EOF()  

    *  

    ** CHANGE OF PROGRESS  

  *

```

```

IF pct > 0 .AND. pct <> wspct
  IF pct < wspct
    comments = '*** CURRENT PCT LESS THAN PREVIOUS PCT'
  ELSE
    comments = ''
  ENDIF
  DO printact
  SELECT 1
  comments = ''
  DO printact
  Line = Line + 1
  num = num +1
  SKIP 1
  LOOP
ENDIF
IF DTOC(af) <> DTOC(wsaf) .OR. DTOC(as) <> DTOC(wsas) .AND. pct > 0
  DO printact
  SELECT 1
  DO printact
  Line = Line + 1
  num = num +1
  SKIP 1
  LOOP
ENDIF
ENDIF
SELECT 1
SKIP 1
ENDDO
*
Line = Line + 2
?
? 'THE TOTAL NUMBER OF ACTIVITIES FOUND = '+STR(num,3)
*DO TOTOP
DO closefil
RETURN
*: EOF: REPW.PRG
^Z

```

```

*:*****  

*: Program: REPX.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89      22:44  

*: Called by: MAINX.PRG  

*: Call: OPENFILS      (procedure in PROCS.PRG)  

*:       : TITLE1.PRG  

*:       : PRINTPRE.PRG  

*:       : CLOSEFIL      (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****  

DO openfiles  

*  

** INITIALIZE AND PRINT TITLE  

*  

CLEAR  

@ 10,10 SAY "RUNNING REPORT X -- FIND LOGIC CHANGES"  

  

SELECT 1  

GOTO TOP  

SELECT 2  

GOTO TOP  

SELECT 5  

GOTO TOP  

SELECT 6  

GOTO TOP  

*  

PUBLIC wsact1,wssucl,printy,recnum  

STORE SPACE(10) TO wsact,wssuc  

STORE SPACE(12) TO wsact1,wssucl  

STORE 0 TO Line,num,page,printy,recnum,rectemp  

STORE '7' TO rpt1  

DO title1  

Line = Line - 1  

*  

SELECT 2  

DO WHILE .NOT. EOF()  

  recnum = RECNO()  

  printy = 0  

  STORE act TO wsact  

  wsact1 = 'A' + act + 'A'  

*  

** CHECK IF THE ACTIVITY EXISTES IN PREVIOUS FILE  

*  

SELECT 1

```

```

FIND &wsact
IF EOF()           && THIS IS A NEW ACTIVITY IN CURRENT FILE
  SELECT 2
  SKIP 1           && SKIP THIS ACTIVITY
  LOOP
ENDIF
*
** COMPARE RELATION IN BOTH FILES
*
SELECT 6
FIND &wsact
IF .NOT. EOF()
  DO WHILE act = SUBSTR(wsact1,2,10)
    STORE suc TO wssuc
    wsac1 = 'B' + suc + 'B'
    wsactsuc= act + suc
    STORE lag TO wslag
    STORE rel TO wsrel
  *
  SELECT 5
  FIND &wsactsuc
  IF EOF()
    Line = Line + 1           && RELATION ADDED IN CURRENT LOGIC
    SELECT 2
    FIND &wsact
    DO printpre
    FIND &wssuc
    ?? " "+act+" "+TITLE
    SELECT 1
    FIND &wssuc
    IF EOF()
      SELECT 6
      ?? rel+STR(lag,6)+" SUCCESSOR IS NEW ACTIVITY"
      SELECT 1
      FIND &wsact
      DO printpre
    ELSE
      SELECT 6
      ?? rel+STR(lag,6)+" RELATION ADDED"
      SELECT 1
      FIND &wsact
      DO printpre
      SELECT 1
      FIND &wssuc
      ?? " "+act+" "+TITLE
    ENDIF
  *
ELSE                      && SUCCESSOR EXISTS, CHECK FOR
  DO CASE                  && RELATION OR LAG CHANGED
    CASE rel <> wsrel
      SELECT 5
      rectemp = RECNO()

```

```

GOTO RECHNO() + 1
IF act = SUBSTR(wsact1,2,10) .AND. suc = SUBSTR(wssuc1,2,10) .AND. rel = warel
  IF lag <> wslag
    Line = Line + 1      /* CHECK FOR MULTIPLE RELATIONS
    SELECT 2
    FIND &wsact
    DO printpre
    FIND &wssuc
    ?? " "+act+" "+TITLE
    SELECT 6
    ?? rel+STR(lag,6)+" LAG CHANGED"
    SELECT 1
    FIND &wsact
    DO printpre
    FIND &wssuc
    ?? " "+act+" "+TITLE
    SELECT 5
    ?? rel+STR(lag,6)
  ENDIF
  SELECT 6
  SKIP
  LOOP
ENDIF
GOTO rectemp
*
Line = Line + 1
SELECT 2
FIND &wsact
DO printpre
FIND &wssuc
?? " "+act+" "+TITLE
SELECT 6
?? rel+STR(lag,6)+" RELATION CHANGED"
*
SELECT 1
FIND &wsact
DO printpre
FIND &wssuc
?? " "+act+" "+TITLE
SELECT 5
?? rel+STR(lag,6)
*
CASE lag <> wslag
  Line = Line + 1
  SELECT 2
  FIND &wsact
  DO printpre
  FIND &wssuc
  ?? " "+act+" "+TITLE
  SELECT 6
  ?? rel+STR(lag,6)+" LAG CHANGED"
*

```

```

SELECT 1
FIND &wsact
DO printpre
FIND &wssuc
?? " "+act+" "+TITLE
SELECT 5
?? rel+STR(lag,6)
*
ENDCASE
ENDIF
SELECT 6
SKIP
LOOP
ENDDO
ENDIF
*
SELECT 5
FIND &wsact
IF .NOT. EOF()
DO WHILE act = SUBSTR(wsact1,2,10)
STORE suc TO wssuc
wsactsuc= act + suc
SELECT 6
FIND &wsactsuc
IF EOF()
SELECT 2
FIND &wssuc
IF EOF()          && IF SUCCESSOR DOES NOT EXIST
Line = Line + 1      && IN CURRENT FILE, THE SUCCESSOR
SELECT 2          && IS DELETED IN CURRENT FILE
FIND &wsact
DO printpre
SELECT 1
FIND &wsact
DO printpre
FIND &wssuc
?? " "+act+" "+TITLE
SELECT 5
?? rel+STR(lag,6)+" SUCCESSOR IS DELETED ACTIVITY"
ELSE:
Line = Line + 1
SELECT 2
FIND &wsact
DO printpre
FIND &wssuc
?? " "+act+" "+TITLE
SELECT 6
?? rel+STR(lag,6)+" RELATION DELETED"
*
SELECT 1
FIND &wsact
DO printpre

```

```
FIND &WSSUC
?? "+FACT+" "+TITLE
SELECT 5
?? REL+STR(LAG,6)
ENDIF
ENDIF
SELECT 5
SKIP
LOOP
ENDDO
ENDIF
*
SELECT 2
GOTO RECNUM
SKIP
LOOP
ENDDO
Line = Line + 2
?
? '** END OF LOGIC COMPARISON **'
DO CLOSEFILE
RETURN
*: EOF: REPX.PRG
^Z
```

```

*:*****  

*:;  

*:      Program: QUIT.PRG  

*:;  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/20/89      14:51  

*:;  

*:      Called by: REPN1.PRG  

*:              : KEYPRESS.PRG  

*:;  

*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e  

*:*****  

** QUIT.PRG - Clean up all files, Quit Pro-Man;  

    SET COLOR TO &menucol  

CLEAR  

CLOSE ALL  

CLEAR ALL  

* Erase all Pro-Man .NDX files (but don't use DOS wildcard)  

SET CONSOLE OFF  

SET TALK OFF  

SET ECHO OFF  

SET SAFETY OFF  

IF FILE('pssr_act.ndx')  

    ERASE pssr_act.ndx  

ENDIF  

IF FILE('schd_act.ndx')  

    ERASE schd_act.ndx  

ENDIF  

IF FILE('subm_nas.ndx')  

    ERASE subm_nas.ndx  

ENDIF  

IF FILE('tempndx.ndx')  

    ERASE tempndx.ndx  

ENDIF  

IF FILE('schd_stn.ndx')  

    ERASE schd_stn.ndx  

ENDIF  

IF FILE('schd_seq.ndx')  

    ERASE schd_seq.ndx  

ENDIF  

IF FILE('sch2_act.ndx')  

    ERASE sch2_act.ndx  

ENDIF  

IF FILE('sch1_act.ndx')  

    ERASE sch1_act.ndx  

ENDIF  

IF FILE('succ_ndx.ndx')  

    ERASE succ_ndx.ndx  

ENDIF  

IF FILE('mrgndx.ndx')

```

```
ERASE mrgndx.ndx
ENDIF
IF FILE('schd_seq.ndx')
    ERASE schd_seq.ndx
ENDIF
*85,10 SAY 'NOW EXITING PROMAN'
SET COLOR TO
QUIT
*: EOF: QUIT.PRG
^Z
```

```
*:  
*:      Program: REPN.PRG  
*:      System: PROJECT MANAGEMENT TOCLS  
*:      Author: ASC/RCE and USA-CERL  
*:      Copyright (c) 1989, USA-CERL/ASC/RCE  
*:      Last modified: 04/05/89      0:01  
*:      Called by: REPN1.PRG  
*:      Calls: ACTST      (procedure in PROCS.PRG)  
*:              : CODEST      (procedure in PROCS.PRG)  
*:              : PRECST      (procedure in PROCS.PRG)  
*:              : PROGST      (procedure in PROCS.PRG)  
*:      Uses: DATAEXCH.DBF  
*:              : &ASCFILE  
*:              : SCHDINFO.DBF  
*:              : &SCHDFIL  
*:              : SUCINFO.DBF  
*:              : &SUCFIL  
*:      Indexes: TEMPNDX.NDX  
*:      Documented 04/27/89 at 10:40          SNAP! version 3.12e  
*:*****  
SET BELL OFF  
SET TALK OFF  
CLEAR  
@ 10,10 SAY '**** CONVERTING '+ asciiifile+' TO DATABASE FORMAT ****'  
* Begin taking data from asciifile  
  
SELECT F  && Set up for reading of Data Exchange Standard into fields  
USE dataexch    && dataexch.DBF  
ZAP  
APPEND FROM &asciifile SDF  
SELECT G  
Crea &schdfil FROM schdinfo  
SELECT H  
Crea &sucfil FROM sucinfo  
  
SELECT F  
GO TOP  
DO WHILE .NOT. EOF()  
  
IF SUBSTR(ALL,1,4)='PROJ'    && 1 required record in file  
  datadate=SUBSTR(ALL, 6, 1)  
  projid=SUBSTR(ALL, 14, 4)  
  projname=SUBSTR(ALL, 20, 48)  
  contraname=SUBSTR(ALL, 69, 36)  
  arrow_prec=SUBSTR(ALL, 106, 1)
```

```

    proj_userid=SUBSTR(ALL,108,35)
ENDIF

SKIP

N=1
DO WHILE SUBSTR(ALL,1,4)='CLDR'.AND.N<6    && 1 required & >=1 optional records
    calenocode=&n=SUBSTR(ALL, 6,4)
    workdays=&n=SUBSTR(ALL,11,7)
    calendscr=&n=SUBSTR(ALL,19,30)
    SKIP
    N=N+1
ENDDO

SKIP

IF SUBSTR(ALL,1,4)='HOLI'                  && 1 required record
    hcalenocode=SUBSTR(ALL, 6,4)
    N=1
    m=11
    DO WHILE N<14           && Up to 14 holidays
        holiday=&n=SUBSTR(ALL,m,7)
        N=N+1
        m=m+8
        IF holiday=&n=' '
            EXIT
        ENDIF
    ENDDO
ENDIF

SKIP

DO WHILE SUBSTR(ALL,1,4)='ACTV'      && Strip off all Activity data
    activ_id    = SUBSTR(ALL,6,10)      && 4 required fields
    activ_descr = SUBSTR(ALL,17,50)
    activ_dur   = SUBSTR(ALL,68,3)
    activ_cost  = SUBSTR(ALL,72,8)
    IF SUBSTR(ALL,81,7)<>' '          && 6 optional fields
        start_net  = SUBSTR(ALL,81,7)
    ENDIF
    IF SUBSTR(ALL,89,7)<>' '
        finish_nlt = SUBSTR(ALL,89,7)
    ENDIF
    IF SUBSTR(ALL,97,8)<>' '
        unitprice  = SUBSTR(ALL,97,8)
    ENDIF
    IF SUBSTR(ALL,106,7)<>' '
        esti_units = SUBSTR(ALL,106,7)
    ENDIF
    IF SUBSTR(ALL,114,4)<>' '
        unit_descr = SUBSTR(ALL,114,4)
    ENDIF

```

```

IF SUBSTR(ALL,119,4)<>' '
    acalencode = SUBSTR(ALL,119,4)
ENDIF
DO actst
SKIP
ENDDO

SELECT G
INDEX ON act TO tempndx
SELECT F

DO WHILE ((SUBSTR(ALL,1,4)='CODE') .AND. (.NOT. EOF()))
    activ_id      = SUBSTR(ALL,6,10)
    responsibl   = SUBSTR(ALL,17,4)
    modnum       = SUBSTR(ALL,22,5)
    workarea     = SUBSTR(ALL,28,4)
    hamact       = SUBSTR(ALL,33,1)
    breakdown   = SUBSTR(ALL,35,12)
    weathers     = SUBSTR(ALL,48,1)
    act_type     = SUBSTR(ALL,50,4)
    spec         = SUBSTR(ALL,55,3)
    build        = SUBSTR(ALL,64,7)
    calcd        = SUBSTR(ALL,72,4)
    usrdcode     = SUBSTR(ALL,77,56)
    DO codest
    SKIP + 1
ENDDO

DO WHILE ((SUBSTR(ALL,1,4)='LIBR')           .AND. (.NOT. EOF()))
    resid        = SUBSTR(ALL,6,4)
    descr        = SUBSTR(ALL,11,30)
    unitpr       = SUBSTR(ALL,42,8)
    unitdes     = SUBSTR(ALL,51,4)
    totestquan  = SUBSTR(ALL,56,7)
    quanperiod   = SUBSTR(ALL,64,7)
    quantodate  = SUBSTR(ALL,72,7)
    stdate       = SUBSTR(ALL,80,7)
    enddate      = SUBSTR(ALL,88,7)
    SKIP + 1
ENDDO

DO WHILE ((SUBSTR(ALL,1,4)='RESC') .AND. (.NOT. EOF()))
    actidres    = SUBSTR(ALL,6,10)
    residres   = SUBSTR(ALL,17,4)
    quanid      = SUBSTR(ALL,23,1)
    estquan     = SUBSTR(ALL,25,5)
    quanperres  = SUBSTR(ALL,31,7)
    quandteres = SUPSTR(ALL,39,7)
    SKIP + 1
ENDDO

DO WHILE ((SUBSTR(ALL,1,4) = 'PREC') .AND. (.NOT. EOF()))

```

```
actidpr    = SUBSTR(6,10)
sucact     = SUBSTR(17,10)
suctype    = SUBSTR(28,1)
lagamt     = SUBSTR(ALL,30,3)
DO precst.
  SKIP + 1
ENDDO

DO WHILE ((SUBSTR(ALL,1,4) = 'PROG') .AND. (.NOT. EOF()))
  actidprog = SUBSTR(ALL,6,10)
  actst     = SUBSTR(ALL,17,7)
  actf      = SUBSTR(ALL,25,7)
  remdur   = SUBSTR(ALL,33,3)
  earned    = SUBSTR(ALL,37,3)
  unitper   = SUBSTR(ALL,41,7)
  unitdate  = SUBSTR(ALL,49,7)
  DO progat
    SKIP + 1
  ENDDO

ENDDO

*: EOF: REPN.PRG
^Z
```

```

*:*****
*:
*:      Program: GETSCHED.PRG
*:
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/16/89      14:11
*:
*:
*:      Called by: MENUTHRE.PRG
*:
*:
*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****
** GETSCHED.PRG - Prompt user for filenames for SCHEDULE.DBF & SUCCESOR.DBF
SET COLOR TO
CLEAR
@0,5 SAY 'Input the entire filename, including filename extension if present, of'
@1,5 SAY 'the Schedule "SCHEDULE.DBF" file to be compared against PSSR and/or'
@2,5 SAY 'Submittal files. In order to run Reports which require the Schedule data.'
@3,5 SAY 'a filename MUST be ~ here. (This file is generated as a .DBF export)'
@4,5 SAY 'file from Primavera, using Export File Report EX-01. For more specific'
@5,5 SAY 'information about the necessary format for this file, please refer to'
@6,5 SAY 'the PROMAN User Manual.)'
DO WHILE .T.
  filsched = filsched + "
@8,1 SAY 'INPUT SCHEDULE "Schedule" FILENAME: ' GET filsched PICTURE '0$500!'
@9,0 CLEAR
READ
filsched=LTRIM(RTRIM(filsched))
IF FILE(filsched)
  @9,15 SAY 'VALID FILENAME'
  EXIT
ELSE
  SET COLOR TO &errcol
  @9,15 SAY 'CANNOT FIND THIS FILENAME - WOULD YOU LIKE TO:'
  SET COLOR TO
  @10,5 SAY ' 1) QUIT THIS PROCESS AND RETURN TO THE MENU?'
  ch=2
  @11,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?      SELECT 1 OR 2 : 'GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
    RETURN
  ENDIF
ENDIF
ENDDO
@12,5 SAY 'Next, input the entire filename, including filename extension if present,' 
@13,5 SAY 'of the Schedule "SUCCESSOR" Dbase file to be compared against PSSR and/or'
@14,5 SAY 'Submittal files. In order to run Reports which require the Schedule data.'
@15,5 SAY 'a filename MUST be input here. (This file is generated as a .DBF export)'
@16,5 SAY 'file from Primavera, using Export File Report EX-02. For more specific'
@17,5 SAY 'information about the necessary format for this file, please refer to'

```

```

@18,5 SAY 'the PROMAN User Manual.)'
DO WHILE .T.
  filsucces = filsucces + "
@20,1 SAY 'INPUT SCHEDULE "Successor" FILENAME: ' GET filsucces PICTURE '@S50E!'
@21,0 CLEAR
READ
filsucces=LTRIM(RTRIM(filsucces))
IF filsucces=filsched
  SET COLOR TO &errcol
  @21,15 SAY 'THIS FILENAME IS THE SAME AS YOUR "Schedule" FILENAME INPUT ABOVE'
  SET COLOR TO
  @22,15 SAY 'RE-INPUT'
  WAIT
  LOOP
ENDIF
IF FILE(filsucces)
  @21,15 SAY 'VALID FILENAME'
  @22,0
  WAIT
  EXIT
ELSE
  SET COLOR TO &errcol
  @21,15 SAY 'CANNOT FIND THIS FILENAME - WOULD YOU LIKE TO:'
  SET COLOR TO
  @22,5 SAY ' 1) QUIT THIS PROCESS AND RETURN TO THE MENU?'
  ch=2
  @23,5 SAY ' 2) INPUT ALTERNATE FILENAME?      SELECT 1 OR 2 : 'GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
    RETURN
  ENDIF
ENDIF
ENDDO
RETURN
*: EOF: GETSCHED.PRG
^Z

```

```

*:*****
*:
*:      Program: SCHEDULE.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      1:12
*:
*:      Called by: MENUTHRE.PRG
*:
*:      Uses: &FILSCHED
*:              : SCHETEMP.DBF
*:              : &FILSCHED.DBF
*:              : FILELIST.DBF
*:
*:      Indexes: SCHD_ACT.NDX
*:
*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****
** SCHEDULE.PRG - Process and clean up SCHEDULE.DBF, the main Primavera export
CLEAR
@5,25 SAY 'PROCESSING SCHEDULE FILE'
@6,22 SAY "PLEASE DON'T TOUCH KEYBOARD"
SELECT D
filsched = UPPER(filsched)
filsucces = UPPER(filsucces)
USE &filsched
IF TYPE("MH_DAY") <> "U"
  CLEAR
  @ 10,10 SAY "The file '"+TRIM(filsched)+"' has been processed already."
  @ 12,10 SAY "Processing of this schedule and the associated successor"
  @ 13,10 SAY "file, '"+TRIM(filsucces)+"' will not be performed."
  @ 15,10 SAY " "
  WAIT
  option = 0
  RETURN
ENDIF
USE
SELECT A
USE schetemp
ZAP
APPEND FROM &filsched
ERASE schd_act.ndx
INDEX ON act TO schd_act
GOTO TOP
SET DELETE ON
N=0
manhrs=0
dollars=0
mact = "ZZZZZZZZZZ"
DO WHILE .NOT. EOF()

```

```

IF mact <> act
  IF TRIM(rut) = '$' .AND. upd <> 0.00  && Move Dollars
    REPLACE dol_day WITH upd
  ELSE
    IF TRIM(rut) = 'MH' .AND. upd <> 0.00  && Move Man Hours
      REPLACE mh_day WITH upd
    ENDIF
  ENDIF
  mact = act
  SKIP
ELSE
  IF TRIM(rut) = '$' .OR. TRIM(rut) = 'MH'
    mrut = TRIM(rut)
    mupd = upd
    DELETE
    SKIP -1
    IF mrut = '$' .AND. mupd <> 0.00
      REPLACE dol_day WITH mupd
    ELSE
      IF mrut = 'MH'
        REPLACE mh_day WITH mupd
      ENDIF
    ENDIF
    SKIP
  ELSE
    DELETE
    SKIP
  ENDIF
ENDIF
ENDDO
PACK
SET DELETE OFF
GOTO TOP
DO WHILE .NOT. EOF()
  N = N + 1
  IF dol_day > 0.00
    dollars = dollars + 1
  ENDIF
  IF mh_day > 0.00
    manhrs = manhrs +1
  ENDIF
  SKIP
ENDDO
ERASE &filsched
GOTO TOP
COPY ALL TO &filsched
GOTO TOP
DELETE ALL
PACK
?
? STR(manhrs,7)+' Activities require some level of Man-Hours.'
? STR(dollars,7)+' Activities require some level of Dollars.'

```

```
? STR(N,7) +' Activities are included in the schedule database.'
```

```
?
```

```
sched_done=.T.
```

```
* start of filelist update
```

```
SELECT A
```

```
USE
```

```
USE filelist
```

```
GOTO TOP
```

```
REPLACE achedule WITH filsched
```

```
REPLACE successor WITH filsucces
```

```
USE
```

```
SELECT D
```

```
USE &filsched
```

```
* end of filelist update
```

```
RETURN
```

```
*: EOF: SCHEDULE.PRG
```

```
^Z
```

```

*:*****
*:
*:      Program: GETSUBMI.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/16/89    14:18
*:
*:      Called by: MENUTHRE.PRG
*:
*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****
** GETSUBMI.PRG - Prompt user for filename for submittal file
CLEAR
@1,5 SAY 'Input the entire filename, including filename extension if present, of'
@2,5 SAY 'the Submittal Dbase file to be compared against Schedule and/or'
@3,5 SAY 'PSSR files. In order to run Reports which require the Submittal data.'
@4,5 SAY 'a filename MUST be input here. (For specific information about the'
@5,5 SAY 'necessary format for this file, please refer to the PROMAN User Manual.)'
DO WHILE .T.
  filsubmi = filsubmi + "
@7,1 SAY 'INPUT SUBMITTAL FILENAME: ' GET filsubmi PICTURE '@S50E1'
@8,0 CLEAR
READ
filsubmi=LTRIM(RTRIM(filsubmi))
IF FILE(filsubmi)
  @9,15 SAY 'VALID FILENAME'
  @10,0
  WAIT
  EXIT
ELSE
  SET COLOR TO &errcol
  @9,15 SAY 'CANNOT FIND THIS FILENAME - WOULD YOU LIKE TO:'
  SET COLOR TO
  @10,5 SAY ' 1) QUIT THIS PROCESS AND RETURN TO THE MENU?'
  ch=2
  @11,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?      SELECT 1 OR 2 : ' GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
    RETURN
  ENDIF
ENDIF
ENDDO
RETURN
*: EOF: GETSUBMI.PRG
^Z

```

```

*:*****
*:
*:      Program: SUBMIT.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*: Last modified: 04/27/89      1:13
*:
*:      Called by: MENUTHRE.PRG
*:
*:      Uses: &FILSUBMI
*:              : FILELIST.DBF
*:
*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e
*:*****
** SUBMIT.PRG - Process submittal file (filename user's choice)
SET COLOR TO &extracol
CLEAR
G2,20 SAY 'PROCESSING SUBMITTAL FILE ....."
G3,25 SAY "PLEASE DON'T TOUCH KEYBOARD"
SELECT E
USE &filsubmi  && Process submittal data
DO WHILE .NOT. EOF()
  IF SUBSTR(nas,5,1)='.'
    REPL nas WITH TRIM(SUBSTR(nas,1,4)+SUBSTR(nas,6,7))
  ENDIF
  SKIP
ENDDO
submi_done=.T.
SET COLOR TO &menucol
* start of file list update
SELECT A
USE
USE filelist
GOTO TOP
REPLACE submittal WITH filsubmi
USE
SELECT E
*end of filelist update
RETURN
*: EOF: SUBMIT.PRG
^Z

```

```

*:*****  

*:  

*:      Program: GETPSSR.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/16/89    14:16  

*:  

*:      Called by: MENUTHRE.PRG  

*:  

*:      Documented 04/27/89 at 10:40           SNAP! version 3.12e  

*:*****  

** GETPSSR.PRG - Get user's filenames for PSSR_PR & PSSR_SC ASCII files  

* Mark McReynolds, 1/3/88  

SET COLOR TO  

CLEAR  

@1,5 SAY 'Input the entire filename, including filename extension if present, of'  

@2,5 SAY 'the PSSR "Procurement" ASCII file to be compared against Schedule and/or'  

@3,5 SAY 'Submittal files. In order to run Reports which require the PSSR data,'  

@4,5 SAY 'a filename MUST be input here. (For specific information about the'  

@5,5 SAY 'necessary format for this file, please refer to the PROMAN User Manual.)'  

DO WHILE .T.  

  filpssr_pr = filpssr_pr + ""  

  @7,1 SAY 'INPUT PSSR "Procurement" FILENAME: ' GET filpssr_pr PICTURE '@S50@!'  

  @8,0 CLEAR  

  READ  

  filpssr_pr=LTRIM(RTRIM(filpssr_pr))  

  IF FILE(filpssr_pr)  

    @9,15 SAY 'VALID FILENAME'  

    EXIT  

  ELSE  

    SET COLOR TO &errcol  

    @9,15 SAY 'CANNOT FIND THIS FILENAME - WOULD YOU LIKE TO:'  

    SET COLOR TO  

    @10,5 SAY ' 1) QUIT THIS PROCESS AND RETURN TO THE MENU?'  

    ch=2  

    @11,5 SAY ' 2) INPUT AN ALTERNATE FILENAME?      SELECT 1 OR 2 : ' GET ch PICTURE '9'  

    READ  

    IF ch=1  

      option = 0  

      RETURN  

    ENDIF  

  ENDIF  

ENDDO  

@11,5 SAY 'Input a PSSR "SubContract" ASCII file? ' GET sc_exist PICTURE 'Y'  

READ  

IF sc_exist  

  @14,5 SAY 'Input the entire filename, including filename extension if present, of'  

  @15,5 SAY 'the PSSR "SubContract" ASCII file to be compared against Schedule and/or'  

  @16,5 SAY 'Submittal files. (For specific information about the necessary format for'  

  @17,5 SAY 'this file, please refer to the PROMAN User Manual.)'

```

```

DO WHILE .T.
filpssr_sc = filpssr_sc + "
Q19,1 SAY 'INPUT PSSR "SubContract" FILENAME:  ' GET filpssr_sc PICTURE '@$50@!'
Q20,0 CLEAR
READ
filpssr_sc=LTRIM(RTRIM(filpssr_sc))
IF filpssr_sc=filpssr_pr
  SET COLOR TO &errcol
  Q20,15 SAY 'THIS FILENAME IS THE SAME AS YOUR "Procurement" PSSR FILENAME.'
  Q21,15 SAY 'INPUT AGAIN.'
  SET COLOR TO
  LOOP
ENDIF
IF FILE(filpssr_sc)
  Q20,15 SAY 'VALID FILENAME'
  Q21,0
  WAIT
  EXIT
ELSE
  SET COLOR TO &errcol
  Q20,15 SAY 'CANNOT FIND THIS FILENAME - WOULD YOU LIKE TO:'
  SET COLOR TO
  Q21,5 SAY ' 1) QUIT THIS PROCESS AND RETURN TO THE MENU?'
  ch=2
  Q22,5 SAY ' 2) INPUT ALTERNATE FILENAME?      SELECT 1 OR 2 : 'GET ch PICTURE '9'
  READ
  IF ch=1
    option = 0
    RETURN
  ENDIF
ENDIF
ENDDO
ENDIF
RETURN
*: EOF: GETPSSR.PRG
^Z

```

```

*:*****  

*:  

*:      Program: PSSR.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      1:22  

*:  

*:      Called by: MENUTHRE.PRG  

*:  

*:      Uses: PSSR_PR.DBF  

*:            : AA.DBF  

*:            : &FILPSSR_PR  

*:            : PSSR_SC.DBF  

*:            : &FILPSSR_SC  

*:            : FILELIST.DBF  

*:  

*:      Documented 04/27/89 at 10:40      SNAP! version 3.12e  

*:*****  

** PSSR.PRG - Creates PSSR_PR.DBF from PSSRS.ART (or user's choice) ASCII  

* file, and PSSR_SC.DBF from PSSRR.ART (or user's choice) ASCII file, merges  

SET COLOR TO &outputcol  

CLEAR  

Q5,9 SAY 'CONVERTING PSSR "Procurement" FILE FROM ASCII TO DBASE FORMAT'  

Q6,25 SAY "PLEASE DON'T TOUCH KEYBOARD"  

SELECT B  

USE pssr_pr  

ZAP  

SELECT A  

USE aa  

ZAP  

APPEND FROM &filpssr_pr SDF  

GO TOP  

mprrepdate=SPACE(13)  

mprreptime=SPACE(5)  

mprrepjob=SPACE(6)  

mprendper=SPACE(8)  

SKIP 7  

IF SUBSTR(ALL,120,1)$'0123456789'  

    mprrepdate=SUBSTR(ALL,120,7)  && Get the date report was printed  

    mprreptime=SUBSTR(ALL,128,5)  && Get the time report was printed  

ENDIF  

SKIP 2  

IF SUBSTR(ALL,1,3)='JOB'  

    mprrepjob=SUBSTR(ALL,4,6)  && Get the report job number  

ENDIF  

SKIP 2  

IF SUBSTR(ALL,119,6)='ENDING' .AND. SUBSTR(ALL,126,1)$'0123456789'  

    mpreadper=SUBSTR(ALL,126,7)  && Get ending period of these activities  

ENDIF  

D=' '

```

```

DO WHILE .NOT. EOF()
  IF SUBSTR(ALL,1,1) $'0123456789'
    SELECT B   && pssr_pr.dbf
    APPEND BLANK
    SELECT A   && aa.dbf
    REPL b->activ_id WITH SUBSTR(ALL,1,12),b->descript WITH SUBSTR(ALL,15,42),:
      b->schedclass WITH SUBSTR(ALL,57,5),b->req_no WITH SUBSTR(ALL,65,5)
    REPL b->p_o_no WITH SUBSTR(ALL,73,7),b->vendorname WITH SUBSTR(ALL,80,14),:
      b->remarks WITH SUBSTR(ALL,94,33)
    SKIP
    REPL b->advertize WITH SUBSTR(ALL,18,7),b->reqtoproc WITH SUBSTR(ALL,27,7),:
      b->bids_out WITH SUBSTR(ALL,36,7),b->bids_in WITH SUBSTR(ALL,45,7)
    REPL b->kebid_eval WITH SUBSTR(ALL,54,7),b->cебid_eval WITH SUBSTR(ALL,63,7),:
      b->award_p_o WITH SUBSTR(ALL,72,7),b->ce_submit WITH SUBSTR(ALL,81,7)
    REPL b->ce_subm_ap WITH SUBSTR(ALL,90,7),b->fab_ship WITH SUBSTR(ALL,99,7),:
      b->arrivesite WITH SUBSTR(ALL,108,9),b->required WITH SUBSTR(ALL,126,7)
    SKIP  && The next REPLACE gets the WORKING DAYS required for each activity
    REPL b->dadvertize WITH SUBSTR(ALL,22,3),b->dreqtoproc WITH SUBSTR(ALL,31,3),:
      b->dbids_out WITH SUBSTR(ALL,40,3),b->dbids_in WITH SUBSTR(ALL,49,3)
    REPL b->dkebid_eva WITH SUBSTR(ALL,58,3),b->dcebaid_eva WITH SUBSTR(ALL,67,3),:
      b->daward_p_o WITH SUBSTR(ALL,76,3),b->dce_submit WITH SUBSTR(ALL,85,3)
    REPL b->dce_subm_a WITH SUBSTR(ALL,94,3),b->dfab_ship WITH SUBSTR(ALL,103,3),:
      b->darrivesit WITH SUBSTR(ALL,112,3),b->drequired WITH SUBSTR(ALL,129,4)
  ENDIF
  SKIP
ENDDO
ZAP  && aa.dbf
IF sc_exist  && Process pssr_sc.dbf IF user has selected this file.
@8,9 SAY 'CONVERTING PSSR "Subcontract" FILE FROM ASCII TO DBASE FORMAT'
@9,25 SAY "PLEASE DON'T TOUCH KEYBOARD"
SELECT C
USE pssr_sc
ZAP
SELECT A  && aa.dbf
APPEND FROM &filpssr_sc SDF
GO TOP
mscrepdate=SPACE(13)
mscreptime=SPACE(5)
mscrepjob=SPACE(6)
mscendper=SPACE(8)
SKIP 7
IF SUBSTR(ALL,120,1)$'0123456789'
  mscrepdate=SUBSTR(ALL,120,7)  && Get the date report was printed
  mscreptime=SUBSTR(ALL,128,5)  && Get the time report was printed
ENDIF
SKIP 2
IF SUBSTR(ALL,1,3)='JOB'
  mscrepjob=SUBSTR(ALL,4,6)  && Get the report job number
ENDIF
SKIP 2
IF SUBSTR(ALL,119,6)='ENDTNG' .AND. SUBSTR(ALL,126,1)$'0123456789'
  mscendper=SUBSTR(ALL,126,7)  && Get ending period of these activities

```

```

ENDIF
DO WHILE .NOT. EOF()
  IF SUBSTR(ALL,1,1) $'0123456789'
    SELECT C   && pssr_sc.dbf
    APPEND BLANK
    SELECT A
    REPL c->activ_id WITH SUBSTR(ALL,1,12),c->descript WITH SUBSTR(ALL,15,42),;
      c->schedclass WITH SUBSTR(ALL,57,5),c->req_no WITH SUBSTR(ALL,65,5)
    REPL c->p_o_no WITH SUBSTR(ALL,73,7),c->vendorname WITH SUBSTR(ALL,80,14),;
      c->remarks WITH SUBSTR(ALL,94,32)
    SKIP
    REPL c->advertize WITH SUBSTR(ALL,18,7),c->pr_bid_doc WITH SUBSTR(ALL,27,7),;
      c->bids_out WITH SUBSTR(ALL,36,7),c->bids_in WITH SUBSTR(ALL,45,7)
    REPL c->bid_eval WITH SUBSTR(ALL,54,7),c->ce_approve WITH SUBSTR(ALL,63,7),;
      c->award_cont WITH SUBSTR(ALL,72,7),c->mobilize_c WITH SUBSTR(ALL,81,7)
    REPL c->required WITH SUBSTR(ALL,126,7)
    SKIP  && The next two REPLA commands gets the DAYS required for each activity
    REPL c->dadvertize WITH SUBSTR(ALL,22,3),c->dprbid_doc WITH SUBSTR(ALL,31,3),;
      c->dbids_out WITH SUBSTR(ALL,40,3),c->dbids_in WITH SUBSTR(ALL,49,4)
    REPL c->dbid_eval WITH SUBSTR(ALL,58,3),c->dce_approv WITH SUBSTR(ALL,67,3),;
      c->daward_con WITH SUBSTR(ALL,76,3),c->dmobilize WITH SUBSTR(ALL,85,3)
    REPL c->drequired WITH SUBSTR(ALL,129,4)
  ENDIF
  SKIP
ENDDO
ZAP  && aa.dbf
Q11,10 SAY 'MERGING PSSR "Procurement" AND "Subcontract" DBASE FILES'
Q12,25 SAY "PLEASE DON'T TOUCH KEYBOARD"
SELECT B  && pssr_pr.dbf
GO BOTTOM
SELECT C  && pssr_sc.dbf
GO TOP
DO WHILE .NOT. EOF()
  SELECT B
  APPEND BLANK
  REPL activ_id WITH c->activ_id, descript WITH c->descript,;
    schedclass WITH c->schedclass, req_no WITH c->req_no
  REPL p_o_no WITH c->p_o_no, vendorname WITH c->vendorname,;
    remarks WITH c->remarks, advertize WITH c->advertize
  REPL reqloproc WITH c->pr_bid_doc, bids_out WITH c->bids_out,;
    bids_in WITH c->bids_in, kebid_eval WITH c->bid_eval
  REPL cebid_eval WITH c->ce_approve, award_p_o WITH c->award_cont,;
    arrivesite WITH c->mobilize_c, required WITH c->required
  REPL dadvertize WITH c->dadvertize, dreqtoproc WITH c->dprbid_doc,;
    dbids_out WITH c->dbids_out, dbids_in WITH c->dbids_in
  REPL dkebid_eva WITH c->dbid_eval, dcebido_eva WITH c->dce_approv,;
    daward_p_o WITH c->daward_con
  REPL darrivesit WITH c->dmobilize, drequired WITH c->drequired
  SKIP
  SELECT C
  SKIP
ENDDO

```

```

ENDIF
SELECT B
GO TOP
DO WHILE .NOT. EOF()
  F=8
  DO WHILE F<20
    fn=FIELD(F)
    om=' '
    IF &fn<>' '
      om=SUBSTR(&fn,3,3)
      REPL &fn WITH (&om+'/'+SUBSTR(&fn,1,2)+'/'+SUBSTR(&fn,6,2))
    ENDIF
    F=F+1
  ENDDO
  SKIP
ENDDO
SET COLOR TO &menucol
CLEAR
pssr_done=.T.
* Start of filelist update
SELECT A
USE
USE filelist
GOTO TOP
REPLACE pssr_proc WITH UPPER(filpssr_pr)
IF sc_exist
  REPLACE pssr_subs WITH UPPER(filpssr_sc)
ENDIF
USE
* End of filelist update
SELECT B
RETURN
*: EOF: PSSR.PRG
^Z

```

```
*****  
*: Program: PAR_E.PRG  
*:  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/26/89 15:56  
*: Called by: SETPARMS.PRG  
*: Documented 04/27/89 at 10:40 SNAP! version 3.12e  
*****  
* Parameter setup for report e
```

CLEAR

```
@1,21 SAY 'REPORT E: PSSR/SCHEDULE/SUBMITTAL COMPARISON 1'  
@3,5 SAY ' Report E identifies Activities which have matching data in the'  
@4,5 SAY 'PSSR and Submittal files but have mismatching submittal dates, and'  
@5,5 SAY 'have less than a specified quantity of Float. By default this Float'  
@6,5 SAY 'is set at 10 days.'  
@8,5 SAY ' However, the user can alternatively supply his own value for'  
@9,5 SAY 'the Float. If you wish to accept the default, simply press <Enter>'  
@10,5 SAY 'following the prompt. If you would rather supply your own value,'  
@11,5 SAY 'type it in and press <Enter>.'  
@13,10 SAY 'Float Value in Working Days > ' GET refloat PICTURE '999'
```

READ

RETURN

\*: EOF: PAR\_E.PRG

^Z

```
*:*****  
*: Program: PAR_F.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/26/89 15:56  
*: Called by: SETPARMS.PRG  
*: Documented 04/27/89 at 10:40 SNAP! version 3.12e  
*:*****  
* Parameter setup for report f
```

CLEAR

```
Q1,21 SAY 'REPORT F: PSSR/SCHEDULE/SUBMITTAL COMPARISON 2'  
Q3,5 SAY ' Report F identifies Activities which have matching data in the'  
Q4,5 SAY 'PSSR and Submittal files but have mismatching submittal dates, and'  
Q5,5 SAY 'have a time difference between their PSSR submittal and approval dates'  
Q6,5 SAY 'of less than a specified percentage of their Float. By default this'  
Q7,5 SAY 'Float percentage is set at 20%.'  
Q9,5 SAY ' However, the user can alternatively supply his own value for'  
Q10,5 SAY 'this Float percentage. If you wish to accept the default, simply press'  
Q11,5 SAY '<Enter> following the prompt. If you would rather supply your own'  
Q12,5 SAY 'percentage, type it in and press <Enter>.'  
Q14,10 SAY 'Percentage of Float (as described above) > ' GET rfpercent PICTURE '999%'
```

READ

RETURN

```
*: EOF: PAR_F.PRG
```

^Z

```

*:*****  

*: Program: PAR_G.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 15:57  

*: Called by: SETPARMS.PRG  

*: Documented 04/27/89 at 10:41 SNAP! version 3.12e  

*:*****  

* Get parameters for Report G

CLEAR
@1,17 SAY 'REPORT G: RESOURCE LOADING ANALYSIS - MANPOWER VARIANCE'
@3,5 SAY 'Report G analyzes those Activities requiring more than a particular'
@4,5 SAY 'number of men per day. Input the number of men per day per Activity ABOVE'
@5,5 SAY 'which to search (use a number between 0 and 9999), or press <Enter> to use'
DO WHILE .T.
  @6,5 SAY 'the default of 20 men per day per Activity: ' GET rgmn PICTURE '9999'
  READ
  IF rgmn<0.OR.rgmn>9999
    SET COLOR TO $errcol
    @8,10 SAY 'CHOICE IS OUT OF RANGE - PLEASE CHOOSE A NUMBER BETWEEN 0 AND 9,999.'
    SET COLOR TO
  ELSE
    @8,10 SAY 'Valid choice.'
    EXIT
  ENDIF
ENDDO

@10,5 SAY 'Next, input the threshold of standard deviations above which to analyze'
@11,5 SAY 'Activities (use a number between 0 and 99.99, or press <Enter> to use the'
DO WHILE .T.

  @12,5 SAY 'the default of 1.00: ' GET rgsd PICTURE '99.99'
  READ
  IF rgsd<0.OR.rgsd>99.99
    SET COLOR TO $errcol
    @14,10 SAY 'CHOICE IS OUT OF RANGE - PLEASE CHOOSE A NUMBER BETWEEN 0 AND 99.99.'
    SET COLOR TO
  ELSE
    @14,10 SAY 'Valid choice.'
    EXIT
  ENDIF
ENDDO

CLEAR
@1,17 SAY 'REPORT G: RESOURCE LOADING ANALYSIS - MANPOWER VARIANCE'
@3,5 SAY 'Finally, you may choose one of the following Activity Codes or'

```

```
Q4,5 SAY 'Activity ID Codes to establish the grouping for Mean and Standard'
Q5,5 SAY 'Deviation calculations.'
Q7,10 SAY '1    STND Activity ID Code'
Q8,10 SAY '2    RESPONSibility Activity Code'
Q9,10 SAY '3    WAD Activity Code'
Q10,10 SAY '4    SORT Details Activity Code'
Q11,10 SAY '5    SUBcontractS Activity Code'
Q12,10 SAY '6    AREA Activity ID Code'
Q13,10 SAY '7    Sub-AREA Activity ID CCode'
Q14,10 SAY '8    TYPE of Work Activity ID Code'
DO WHILE .T.
  Q16,20 SAY 'Enter the number for one Code : ' GET rgcode PICTURE '9'
  READ
  IF .NOT.rgcode$'12345678'
    SET COLOR TO &errcol
    Q18,20 SAY 'WRONG CHOICE -  SELECT AGAIN'
    SET COLOR TO
  ELSE
    Q18,20 SAY 'Valid Code'
    EXIT
  ENDIF
ENDDO

RETURN
*: EOF: PAR_G.PRG
^Z
```

```
*****  

*: Program: PAR_H.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 15:57  

*: Called by: SETPARMS.PRG  

*: Documented 04/27/89 at 10:41 SNAP! version 3.12e  

*****
```

\* Get Parameters for Report H

CLEAR

```
G1,19 SAY 'REPORT H: RESOURCE LOADING ANALYSIS - FUNDING VARIANCE'  

G3,5 SAY ' Report H analyzes those Activities requiring more than a particular'  

G4,5 SAY 'number of dollars per day. Input the number of dollars per day per'  

G5,5 SAY 'Activity ABOVE which to search (use a number between $0 and $99,999), or'  

DO WHILE .T.  

  G6,5 SAY 'press <Enter> to use the default of $2000 per day per Activity: ' GET rhdl PICTURE  

'99,999'  

  READ  

  IF rhdl<0.OR.rhdl>99999  

    SET COLOR TO &errcol  

    G8,10 SAY 'CHOICE IS OUT OF RANGE - PLEASE CHOOSE A NUMBER BETWEEN 0 AND 99,999'  

    SET COLOR TO  

  ELSE  

    G8,10 SAY 'Valid choice.'  

    EXIT  

  ENDIF  

ENDDO  

G10,5 SAY 'Next, input the threshold of standard deviations above which to analyze'  

G11,5 SAY 'Activities (use a number between 0 and 99.99, or press <Enter> to use the'  

DO WHILE .T.  

  G12,5 SAY 'the default of 1: ' GET rhsd PICTURE '99.99'  

  READ  

  IF rhsd < 0.0 .OR. rhsd > 99.99  

    SET COLOR TO &errcol  

    G14,10 SAY 'YOUR CHOICE IS OUT OF RANGE. PLEASE CHOOSE A NUMBER BETWEEN 0 AND 99.99.'  

    SET COLOR TO  

  ELSE  

    G14,10 SAY 'Valid choice.'  

    EXIT  

  ENDIF  

ENDDO  

CLEAR  

G1,19 SAY 'REPORT H: RESOURCE LOADING ANALYSIS - FUNDING VARIANCE'
```

```

@3,5 SAY ' Finally, you may choose one of the following Activity Codes or'
@4,5 SAY 'Activity ID Codes to establish the grouping for Mean and Standard'
@5,5 SAY 'Deviation calculations.'
@7,10 SAY '1 STND Activity ID Code'
@8,10 SAY '2 RESPonsibility Activity Code'
@9,10 SAY '3 WAD Activity Code'
@10,10 SAY '4 SORT Details Activity Code'
@11,10 SAY '5 SUBcontracts Activity Code'
@12,10 SAY '6 AREA Activity ID Code'
@13,10 SAY '7 Sub-ARea Activity ID C0de'
@14,10 SAY '8 TYPE of Work Activity ID Code'

DO WHILE .T.

@16,20 SAY 'Enter the number for one Code : ' GET rhcode PICTURE '9'
READ
IF .NOT.rhcode$'12345678'
  SET COLOR TO &errcol
  @18,20 SAY 'WRONG CHOICE - SELECT AGAIN'
  SET COLOR TO
  rhcode='1'
ELSE
  @18,20 SAY 'Valid Code'
  EXIT
ENDIF
ENDDO

RETURN
*: EOF: PAR_H.PRG
^Z

```

```
*:*****  

*: Program: PAR_I.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 15:57  

*: Called by: SETPARMS.PRG  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  

*:*****
```

\* Get Parameters for Report I

CLEAR

```
Q1,20 SAY 'REPORT I: RESOURCE LOADING ANALYSIS - TIME VARIANCE'  

Q3,5 SAY 'Report I analyzes those Activities requiring more than a particular'  

Q4,5 SAY 'number days to complete. Input the number of days per Activity ABOVE'  

Q5,5 SAY 'which to search (use a number between 1 and 9,999), or press <Enter>'  

DO WHILE .T.  

  Q6,5 SAY 'to use the default of 40 days per Activity: ' GET ridy PICTURE '9,999'  

  READ  

  IF ridy<1.OR.ridy>9999  

    SET COLOR TO &errcol  

    Q8,10 SAY 'YOUR CHOICE IS OUT OF RANGE. PLEASE CHOOSE A NUMBER BETWEEN 1 AND 9,999.'  

    SET COLOR TO  

  ELSE  

    Q8,10 SAY 'Valid choice.'  

    EXIT  

  ENDIF  

ENDDO  

Q10,5 SAY 'Next, input the threshold of standard deviations above which to analyze'  

Q11,5 SAY 'Activities (use a number between 0 and 99.99, or press <Enter> to use the'  

DO WHILE .T.  

  Q12,5 SAY 'the default of 1: ' GET risd PICTURE '99.99'  

  READ  

  IF risd<0.OR.risd>99.99  

    SET COLOR TO &errcol  

    Q14,10 SAY 'YOUR CHOICE IS OUT OF RANGE. PLEASE CHOOSE A NUMBER BETWEEN 0 AND 99.99.'  

    SET COLOR TO  

  ELSE  

    Q14,10 SAY 'Valid choice.'  

    EXIT  

  ENDIF  

ENDDO  

CLEAR  

Q1,20 SAY 'REPORT I: RESOURCE LOADING ANALYSIS - TIME VARIANCE'  

Q3,5 SAY 'Finally, you may choose one of the following Activity Codes or'  

Q4,5 SAY 'Activity ID Codes to establish the grouping for Mean and Standard'  

Q5,5 SAY 'Deviation calculations.'
```

```
Q7,10 SAY '1    STND Activity ID Code'
Q8,10 SAY '2    RESPONSibility Activity Code'
Q9,10 SAY '3    WAD Activity Code'
Q10,10 SAY '4    SORT Details Activity Code'
Q11,10 SAY '5    SUBcontractS Activity Code'
Q12,10 SAY '6    AREA Activity ID Code'
Q13,10 SAY '7    SUB-AREA Activity ID CCode'
Q14,10 SAY '8    TYPE of Work Activity ID Code'

DO WHILE .T.
  Q16,20 SAY 'Enter the number for one Code : ' GET ricode PICTURE '9'
  READ
  IF .NOT.ricode$'12345678'
    SET COLOR TO &errcol
    Q18,20 SAY 'WRONG CHOICE -  SELECT AGAIN'
    ricode = '1'
    SET COLOR TO
  ELSE
    Q18,20 SAY 'Valid Code'
    EXIT
  ENDIF
ENDDO
RETURN
*: EOF: PAR_I.PRG
^Z
```

```
*:*****  
*: Program: PAR_K.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/26/89 16:00  
*: Called by: SETPARMS.PRG  
*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  
*:*****
```

```
* Get Parameters for Report K
```

```
CLEAR
```

```
@ 1,27 SAY "REPORT K: ACTIVITIES TO CRASH"
```

```
@ 3, 5 SAY "Report K will find the 4 most critical paths in the "
```

```
@ 4, 5 SAY "schedule and print those. Report K will also find those"
```

```
@ 5, 5 SAY "activities which meet certain criteria. It checks for the"
```

```
@ 6, 5 SAY "a certain values of Man Power Loading and Duration."
```

```
@ 14,10 SAY "What value for Man Power Loading to use: " GET mpl PICTURE '999'
```

```
READ
```

```
@ 15,10 SAY "What value for Duration to use: " GET durat PICTURE '99999'
```

```
READ
```

```
@ 16,10 SAY "What value for Relation Duration to use: " GET reldur PICTURE '999'
```

```
READ
```

```
RETURN
```

```
*: EOF: PAR_K.PRG
```

```
^Z
```

```

*:*****  

*: Program: PAR_L.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      6:32  

*: Called by: SETPARMS.PRG  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  

*:*****  

* Routine to Get Parameters for Report L  

  

CLEAR  

@1,29 SAY [REPORT L: DELAYS IN DESIGN]  

@2,4 SAY ""  

TEXT  

Report L compares the loss of Float in Activities involving design  

drawings (or any other user-selected STND Code value), between two  

different Schedule update files. You will need to input two schedule files  

which must have previously been exported from Primavera and processed  

in PROMAN (see user manual for details). Input file names below. Enter  

the earliest schedule file first.  

ENDTEXT  

  

DO WHILE .T.  

  filsch?1 = filsch1 + "          "  

  @10,4 SAY 'Input the filename of your EARLIEST Schedule file: ' GET filsch1  

  READ  

  filsch1=TRIM(filsch1);  

  IF FILE(filsch1)  

    @12,15 SAY 'VALID FILENAME'  

    EXIT  

  ELSE  

    SET COLOR TO &errcol  

    @12,15 SAY 'CANNOT FIND '+TRIM(filsch1)+ ' - WOULD YOU LIKE TO:'  

    SET COLOR TO &menucol  

    @14,15 SAY ' 1) Return to the Main Menu?'  

    ch=2  

    @15,15 SAY ' 2) Input an alternate filename?      Select 1 OR 2 : ' GET ch PICTURE '9'  

    READ  

    IF ch=1  

      option = 0  

      RETURN  

    ENDIF  

    @12,1 CLEAR TO 15,79  

  ENDIF  

ENDDO  

  

DO WHILE .T.

```

```

filsch21 = filsched + "
Q17,4 SAY 'Input the filename of your LATEST Schedule file: 'GET filsch21
READ
filsch21=TRIM(filsch21)
IF FILE(filsch21)
  Q19,15 SAY 'VALID FILENAME'
  EXIT
ELSF
  SET COLOR TO &errcol
  Q19,15 SAY 'CANNOT FIND '+TRIM(filsch21)+ ' - WOULD YOU LIKE TO:'
  SET COLOR TO &menucol
  Q21,15 SAY ' 1) Return to the Main Menu?'
  ch=2
  Q22,15 SAY ' 2) Input an alternate filename?      Select 1 OR 2 : 'GET ch PICTURE '9'
  IF ch=1
    option = 0
    RETURN
  ENDIF
  Q17,1 CLEAR TO 22,79
ENDIF
ENDDO

CLEAR
Q1,29 SAY [REPORT L: DELAYS IN DESIGN]
Q3,5 SAY '      Following is a list of STND Code values present in the two Schedule'
Q4,5 SAY 'files you have entered:'
Q6,10 SAY '01 = General                      11 = Equipment'
Q7,10 SAY '02 = Site Utilities                12 - Furnishings'
Q8,10 SAY '03 = Concrete                     13 - Special Construction'
Q9,10 SAY '04 = Masonry                      14 - Conveying Systems'
Q10,10 SAY '05 = Metal                        15 - Mechanical'
Q11,10 SAY '06 = Wood and Plastics            DR - DRawings'
Q12,10 SAY '07 = Moisture-Thermal Control       PM - Prep Meetings'
Q13,10 SAY '08 = Doors, Windows & Glass        PR - PRocurement'
Q14,10 SAY '09 = Finishes                     SC = SubContract Award'
Q15,10 SAY '10 = Specialties'

DO WHILE .T.
  Q17,5 SAY [You may select the STND code item on which to run Report L, or press]
  Q18,5 SAY [<RETURN> to accept the default Code, 'DR' for Design Drawings: ] GET stnd PICTURE '!!!'
READ
stnd=TRIM(stnd)
IF .NOT.stnd$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC'
  SET COLOR TO &errcol
  Q20,10 SAY 'THAT STND CODE DOES NOT EXIST - PLEASE RE-ENTER.'
  SET COLOR TO &menucol
ELSE
  EXIT
ENDIF
ENDDO

CLEAR

```

```
@1,29 SAY [REPORT L: DELAYS IN DESIGN]
@3,5 SAY 'You may also select the Float threshold value (in days of Float) and'
@4,5 SAY 'percentage of Float lost between Schedule updates, above which to'
@5,5 SAY 'in Report L. Or, you can press <Enter> to accept the default'
@6,10 SAY 'values below.'
@8,10 SAY 'ENTER DAYS OF FLOAT : ' GET ufloat PICTURE '999'
@9,10 SAY 'ENTER PERCENTAGE OF FLOAT : ' GET upercent PICTURE '999%'
READ
upercent = upercent/100    && convert from percentage to decimal equiv.

RETURN
*: EOF: PAR_L.PRG
^Z
```

```

*:*****  

*: Program: REPM_INT.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      6:28  

*: Called by: SETPARMS.PRG  

*: Uses: GRADING1.DBF  

*:           : GRADING2.DBF  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e
*:*****  

** REP_INT.PRG - Get user-specified data for Report M
SET COLOR TO
CLEAR
G1,1 SAY " "
TEXT
REPORT M: CONTRACTOR GRADING

```

The DATA DATE is required to run this program. The schedule data date reflects the effective date of schedule information as computed in primavera.

Please enter the DATA DATE below. The report will not be run without this value. If no value is entered you will be returned to the Main Menu.

Enter the DATA DATE in the form: MM/DD/YY. The slashes will be placed automatically and only a valid date will be accepted.

ENDTEXT

```

DC WHILE .T.
SET CONFIRM ON
317,32 GET datadate
READ
SET CONFIRM OFF
IF DTOC(datadate) = " / / "
@19,10 SAY "Since the data date was left blank, you will be"
@20,10 SAY "returned to the main menu unless you enter an 'N'"
@21,10 SAY "at this time."
@22,10 SAY " "
yn = "Y"
@23,20 SAY "Your choice: " GET yn
READ
yn=UPPER(yn)
IF yn <> "N"
  option = 0
RETURN

```

```
ENDIF
@17,10 CLEAR TO 23,79
ELSE
  EXIT
ENDIF
ENDDO
CLEAR
@1,2 SAY " "
TEXT
REPORT M: CONTRACTOR GRADING
```

Report M enables the user to set scales by which to "grade" or rate the contractor's performance in a number of areas, and then to analyze this performance.

In the following screens, you will be given an opportunity to set various scales by which performance will be judged. The allowable range of threshold values will be shown next to each entry box. The default value is in the box; to accept it simply press the <Enter> key. To choose another value, type that value into the entry box. An entire page be accepted at once by pressing the <PgDn> key.

```
ENDTEXT
WAIT

CLEAR

COPY FILE gradldef.dbf TO grading1.dbf
SELECT F
USE grading1
SET COLOR TO &extracol
@0,25 SAY 'GRADING ITEM 1: SCHEDULE REPORTING'
@1,28 SAY 'Performance Category: Poor'
SET COLOR TO
@3,1 SAY 'A) Between      % and      % of PSSR and submittal (from submittal'
@4,1 SAY ' database) Activities directly correspond to schedule Activities.'
@5,1 SAY 'B) Between      % and      % of procurement Activities in the Schedule'
@6,1 SAY ' are tied into the corresponding PSSR and submittal (from submittal'
@7,1 SAY ' database) items.'
@8,1 SAY 'C) Between      % and      % of construction Activities (i.e.,'
@9,1 SAY ' nonprocurement Activities) have durations exceeding 20 days.'
@10,1 SAY 'D) Between     % and      % of Schedule Activities are coded correctly.'
@3 11 GET a1 PICTURE '999.9'
@3,23 GET a2 PICTURE '999.9'
@5,11 GET b1 PICTURE '999.9'
@5,23 GET b2 PICTURE '999.9'
@8,11 GET c1 PICTURE '999.9'
@8,23 GET c2 PICTURE '999.9'
@10,11 GET d1 PICTURE '999.9'
@10,23 GET d2 PICTURE '999.9'
READ
SKIP
```

SET COLOR TO &extracol  
Q13,26 SAY 'Performance Category: Marginal'  
SET COLOR TO  
Q15,1 SAY 'A) Between % and % of PSSR and submittal (from submittal'  
Q16,1 SAY ' database) Activities directly correspond to schedule Activities.'  
Q17,1 SAY 'B) Between % and % of procurement Activities in the Schedule'  
Q18,1 SAY ' are tied into the corresponding PSSR and submittal (from submittal'  
Q19,1 SAY ' database) items.'  
Q20,1 SAY 'C) Between % and % of construction Activities (i.e.,'  
Q21,1 SAY ' nonprocurement Activities) have durations exceeding 20 days.'  
Q22,1 SAY 'D) Between % and % of Schedule Activities are coded correctly.'  
Q15,11 GET a1 PICTURE '999.9'  
Q15,23 GET a2 PICTURE '999.9'  
Q17,11 GET b1 PICTURE '999.9'  
Q17,23 GET b2 PICTURE '999.9'  
Q20,11 GET c1 PICTURE '999.9'  
Q20,23 GET c2 PICTURE '999.9'  
Q22,11 GET d1 PICTURE '999.9'  
Q22,23 GET d2 PICTURE '999.9'  
READ  
SKIP  
CLEAR  
SET COLOR TO &extracol  
Q0,25 SAY 'GRADING ITEM 1: SCHEDULE REPORTING'  
Q1,25 SAY 'Performance Category: Acceptable'  
SET COLOR TO  
Q3,1 SAY 'A) Between % and % of PSSR and submittal (from submittal'  
Q4,1 SAY ' database) Activities directly correspond to schedule Activities.'  
Q5,1 SAY 'B) Between % and % of procurement Activities in the Schedule'  
Q6,1 SAY ' are tied into the corresponding PSSR and submittal (from submittal'  
Q7,1 SAY ' database) items.'  
Q8,1 SAY 'C) Between % and % of construction Activities (i.e.,'  
Q9,1 SAY ' nonprocurement Activities) have durations exceeding 20 days.'  
Q10,1 SAY 'D) Between % and % of Schedule Activities are coded correctly.'  
Q3,11 GET a1 PICTURE '999.9'  
Q3,23 GET a2 PICTURE '999.9'  
Q5,11 GET b1 PICTURE '999.9'  
Q5,23 GET b2 PICTURE '999.9'  
Q8,11 GET c1 PICTURE '999.9'  
Q8,23 GET c2 PICTURE '999.9'  
Q10,11 GET d1 PICTURE '999.9'  
Q10,23 GET d2 PICTURE '999.9'  
READ  
SKIP  
SET COLOR TO &extracol  
Q13,28 SAY 'Performance Category: Good'  
SET COLOR TO  
Q15,1 SAY 'A) Between % and % of PSSR and submittal (from submittal'  
Q16,1 SAY ' database) Activities directly correspond to schedule Activities.'  
Q17,1 SAY 'B) Between % and % of procurement Activities in the Schedule'  
Q18,1 SAY ' are tied into the corresponding PSSR and submittal (from submittal'  
Q19,1 SAY ' database) items.'

```

@20,1 SAY 'C) Between      % and      % of construction Activities (i.e.,'
@21,1 SAY ' nonprocurement Activities) have durations exceeding 20 days.'
@22,1 SAY 'D) Between      % and      % of Schedule Activities are coded correctly.'
@15,11 GET a1 PICTURE '999.9'
@15,23 GET a2 PICTURE '999.9'
@17,11 GET b1 PICTURE '999.9'
@17,23 GET b2 PICTURE '999.9'
@20,11 GET c1 PICTURE '999.9'
@20,23 GET c2 PICTURE '999.9'
@22,11 GET d1 PICTURE '999.9'
@22,23 GET d2 PICTURE '999.9'

READ
SKIP
CLEAR
SET COLOR TO &extracol
@0,25 SAY 'GRADING ITEM 1: SCHEDULE REPORTING'
@1,26 SAY 'Performance Category: Excellent'
SET COLOR TO
@3,1 SAY 'A) Between      % and      % of PSSR and submittal (from submittal'
@4,1 SAY ' database) Activities directly correspond to schedule Activities.'
@5,1 SAY 'B) Between      % and      % of procurement Activities in the Schedule'
@6,1 SAY ' are tied into the corresponding PSSR and submittal (from submittal'
@7,1 SAY ' database) items.'
@8,1 SAY 'C) Between      % and      % of construction Activities (i.e.,'
@9,1 SAY ' nonprocurement Activities) have durations exceeding 20 days.'
@10,1 SAY 'D) Between      % and      % of Schedule Activities are coded correctly.'
@3,11 GET a1 PICTURE '999.9'
@3,23 GET a2 PICTURE '999.9'
@5,11 GET b1 PICTURE '999.9'
@5,23 GET b2 PICTURE '999.9'
@8,11 GET c1 PICTURE '999.9'
@8,23 GET c2 PICTURE '999.9'
@10,11 GET d1 PICTURE '999.9'
@10,23 GET d2 PICTURE '999.9'

READ
USE
*****
CLEAR
SELECT C
COPY FILE grad2def.dbf TO grading2.dbf
USE grading?
SET COLOR TO &extracol
@0,25 SAY 'GRADING ITEM 2: SCHEDULE PROGRESS'
@1,28 SAY 'Performance Category: Poor'
SET COLOR TO
@3,1 SAY 'A ) There are between      and      specific Activities that have'
@4,1 SAY ' negative Float.'
@5,1 SAY 'B) Between      % and      % of Activities on the four most critical'
@6,1 SAY ' paths (which could have started during this time period), started'
@7,1 SAY ' on their "Early Start Date".'
@8,1 SAY 'C) Between      % and      % of Activities on the four most critical'
@9,1 SAY ' paths (which could have started during this time period), have a "Percent"

```

Q10,1 SAY ' "Complete" value equal to or higher than the porportion to the days'  
Q11,1 SAY ' expended and the days remaining, based on current schedule progress.'  
Q12,1 SAY 'D) Between % and % of Activities have "Actual Start Dates"'  
Q13,1 SAY ' which will reach "Percent Complete" if the "Actual Start Date" is'  
Q14,1 SAY ' over 1 week earlier than the "Data Date".'  
Q15,1 SAY 'E) Between % and % of Activities have "Percent Complete"'  
Q16,1 SAY ' values greater than 0.0% or a "Remaining Duration" with no "Actual"  
Q17,1 SAY ' Start Date'.  
Q18,1 SAY 'F) Between % and % of Activities have a "Remaining Duration"'  
Q19,1 SAY ' greater than their "Original Duration".'

USE grading2

Q3,22 GET a1 PICTURE '999999'  
Q3,33 GET a2 PICTURE '999999'  
Q5,11 GET b1 PICTURE '999.9'  
Q5,23 GET b2 PICTURE '999.9'  
Q8,11 GET c1 PICTURE '999.9'  
Q8,23 GET c2 PICTURE '999.9'  
Q12,11 GET d1 PICTURE '999.9'  
Q12,23 GET d2 PICTURE '999.9'  
Q15,11 GET e1 PICTURE '999.9'  
Q15,23 GET e2 PICTURE '999.9'  
Q18,11 GET f1 PICTURE '999.9'  
Q18,23 GET f2 PICTURE '999.9'

READ

SKIP

CLEAR

SET COLOR TO &extracol

Q0,25 SAY 'GRADING ITEM 2: SCHEDULE PROGRESS'  
Q1,26 SAY 'Performance Category: Marginal'

SET COLOR TO

Q3,1 SAY 'A) There are between and specific Activities that have'  
Q4,1 SAY ' negative Float.'  
Q5,1 SAY 'B) Between % and % of Activities on the four most critical'  
Q6,1 SAY ' paths (which could have started during this time period), started'  
Q7,1 SAY ' on their "Early Start Date".'  
Q8,1 SAY 'C) Between % and % of Activities on the four most critical'  
Q9,1 SAY ' paths (which could have started during this time period), have a "Percent"  
Q10,1 SAY ' "Complete" value equal to or higher than the porportion to the days'  
Q11,1 SAY ' expended and the days remaining, based on current schedule progress.'  
Q12,1 SAY 'D) Between % and % of Activities have "Actual Start Dates"'  
Q13,1 SAY ' which will reach "Percent Complete" if the "Actual Start Date" is'  
Q14,1 SAY ' over 1 week earlier than the "Data Date".'  
Q15,1 SAY 'E) Between % and % of Activities have "Percent Complete"'  
Q16,1 SAY ' values greater than 0.0% or a "Remaining Duration" with no "Actual"  
Q17,1 SAY ' Start Date'.  
Q18,1 SAY 'F) Between % and % of Activities have a "Remaining Duration"'  
Q19,1 SAY ' greater than their "Original Duration".'

Q3,21 GET a1 PICTUREZ '999999'  
Q3,32 GET a2 PICTURE '999999'  
Q5,11 GET b1 PICTURE '999.9'  
Q5,23 GET b2 PICTURE '999.9'  
Q8,11 GET c1 PICTURE '999.9'

```

@8,23 GET c2 PICTURE '999.9'
@12,11 GET d1 PICTURE '999.9'
@12,23 GET d2 PICTURE '999.9'
@15,11 GET e1 PICTURE '999.9'
@15,23 GET e2 PICTURE '999.9'
@18,11 GET f1 PICTURE '999.9'
@18,23 GET f2 PICTURE '999.9'

READ
SKIP
CLEAR

SET COLOR TO &extracol

@0,25 SAY 'GRADING ITEM 2: SCHEDULE PROGRESS'
@1,25 SAY 'Performance Category: Acceptable'
SET COLOR TO

@3,1 SAY 'A) There are between      and      specific Activities that have'
@4,1 SAY ' negative Float.'
@5,1 SAY 'B) Between      % and      % of Activities on the four most critical'
@6,1 SAY ' paths (which could have started during this time period), started'
@7,1 SAY ' on their "Early Start Date".'
@8,1 SAY 'C) Between      % and      % of Activities on the four most critical'
@9,1 SAY ' paths (which could have started during this time period), have a "Percent'
@10,1 SAY ' Complete" value equal to or higher than the proportion to the days'
@11,1 SAY ' expended and the days remaining, based on current schedule progress.'
@12,1 SAY 'D) Between      % and      % of Activities have "Actual Start Dates"'
@13,1 SAY ' which will reach "Percent Complete" if the "Actual Start Date" is'
@14,1 SAY ' over 1 week earlier than the "Data Date".'
@15,1 SAY 'E) Between      % and      % of Activities have "Percent Complete"'
@16,1 SAY ' values greater than 0.0% or a "Remaining Duration" with no "Actual'
@17,1 SAY ' Start Date'.'
@18,1 SAY 'F) Between      % and      % of Activities have a "Remaining Duration"'
@19,1 SAY ' greater than their "Original Duration".'
@3,21 GET a1 PICTURE '999999'
@3,32 GET a2 PICTURE '999999'
@5,11 GET b1 PICTURE '999.9'
@5,23 GET b2 PICTURE '999.9'
@8,11 GET c1 PICTURE '999.9'
@3,23 GET c2 PICTURE '999.9'
@12,11 GET d1 PICTURE '999.9'
@12,23 GET d2 PICTURE '999.9'
@15,11 GET e1 PICTURE '999.9'
@15,23 GET e2 PICTURE '999.9'
@18,11 GET f1 PICTURE '999.9'
@18,23 GET f2 PICTURE '999.9'

READ
SKIP
CLEAR

SET COLOR TO &extracol

@0,25 SAY 'GRADING ITEM 2: SCHEDULE PROGRESS'
@1,28 SAY 'Performance Category: Good'
SET COLOR TO

@3,1 SAY 'A) There are between      and      specific Activities that have'
@4,1 SAY ' negative Float.'

```

Q5,1 SAY 'B) Between        % and        % of Activities on the four most critical'  
Q6,1 SAY ' paths (which could have started during this time period), started'  
Q7,1 SAY ' on their "Early Start Date".'  
Q8,1 SAY 'C) Between        % and        % of Activities on the four most critical'  
Q9,1 SAY ' paths (which could have started during this time period), have a "Percent"  
Q10,1 SAY ' Complete" value equal to or higher than the porportion to the days'  
Q11,1 SAY ' expended and the days remaining, based on current schedule progress.'  
Q12,1 SAY 'D) Between        % and        % of Activities have "Actual Start Dates"  
Q13,1 SAY ' which will reach "Percent Complete" if the "Actual Start Date" is'  
Q14,1 SAY ' over 1 week earlier than the "Data Date".'  
Q15,1 SAY 'E) Between        % and        % of Activities have "Percent Complete"  
Q16,1 SAY ' values greater than 0.0% or a "Remaining Duration" with no "Actual"  
Q17,1 SAY ' Start Date".'  
Q18,1 SAY 'F) Between        % and        % of Activities have a "Remaining Duration"  
Q19,1 SAY ' greater than their "Original Duration".'  
Q3,21 GET a1 PICTURE '999999'  
Q3,32 GET a2 PICTURE '999999'  
Q5,11 GET b1 PICTURE '999.9'  
Q5,23 GET b2 PICTURE '999.9'  
Q8,11 GET c1 PICTURE '999.9'  
Q8,23 GET c2 PICTURE '999.9'  
Q12,11 GET d1 PICTURE '999.9'  
Q12,23 GET d2 PICTURE '999.9'  
Q15,11 GET e1 PICTURE '999.9'  
Q15,23 GET e2 PICTURE '999.9'  
Q18,11 GET f1 PICTURE '999.9'  
Q18,23 GET f2 PICTURE '999.9'  

READ

SKIP

CLEAR

SET COLOR TO &extracol

Q0,25 SAY 'GRADING ITEM 2: SCHEDULE PROGRESS'

Q1,25 SAY 'Performance Category: Excellent'

SET COLOR TO

Q3,1 SAY 'A) There are between        and        specific Activities that have'  
Q4,1 SAY ' negative Float.'

Q5,1 SAY 'B) Between        % and        % of Activities on the four most critical'  
Q6,1 SAY ' paths (which could have started during this time period), started'  
Q7,1 SAY ' on their "Early Start Date".'  
Q8,1 SAY 'C) Between        % and        % of Activities on the four most critical'  
Q9,1 SAY ' paths (which could have started during this time period), have a "Percent"  
Q10,1 SAY ' Complete" value equal to or higher than the porportion to the days'  
Q11,1 SAY ' expended and the days remaining, based on current schedule progress.'  
Q12,1 SAY 'D) Between        % and        % of Activities have "Actual Start Dates"  
Q13,1 SAY ' which will reach "Percent Complete" if the "Actual Start Date" is'  
Q14,1 SAY ' over 1 week earlier than the "Data Date".'  
Q15,1 SAY 'E) Between        % and        % of Activities have "Percent Complete"  
Q16,1 SAY ' values greater than 0.0% or a "Remaining Duration" with no "Actual"  
Q17,1 SAY ' Start Date".'  
Q18,1 SAY 'F) Between        % and        % of Activities have a "Remaining Duration"  
Q19,1 SAY ' greater than their "Original Duration".'  
Q3,21 GET a1 PICTURE '999999'

```
Q3,32 GET a2 PICTURE '999999'
Q5,11 GET b1 PICTURE '999.9'
Q5,23 GET b2 PICTURE '999.9'
Q8,11 GET c1 PICTURE '999.9'
Q8,23 GET c2 PICTURE '999.9'
Q12,11 GET d1 PICTURE '999.9'
Q12,23 GET d2 PICTURE '999.9'
Q15,11 GET e1 PICTURE '999.9'
Q15,23 GET e2 PICTURE '999.9'
Q18,11 GET f1 PICTURE '999.9'
Q18,23 GET f2 PICTURE '999.9'
READ
USE
*
CLEAR
Q8,10 SAY 'Select REPORT M Options Desired:'
DO WHILE .T.
  Q10,10 SAY 'Summary Report (S), Detailed Report (D), or Both (B) > ' GET mch PICTURE '!'
  READ
  Q12,0
  IF .NOT.mch$'SDB'
    SET COLOR TO &errcol
    Q12,20 SAY 'NOT A VALID CHOICE - SELECT S, D OR B'
    SET COLOR TO
  ELSE
    EXIT
  ENDIF
ENDDO
RETURN  && Return to MAIN.PRG
*: EOF: REPM_INT.PRG
^Z
```

```

*:*****  

*: Program: PAR_O.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/21/89      5:06  

*: Called by: SETPARMS.PRG  

*: Calls: CHECK          (procedure in PROCS.PRG)  

*:        : DISPERR        (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e
*:*****  

* Program to Get Parameters for Report O  

DO WHILE .T.  

  CLEAR  

  @ 1,1 SAY " "  

  TEXT  

REPORT O: SLIPPAGE -- REPORT O will look at up to five schedule files and  

compare the values of the float, the original duration, early start date,  

and the productivity rate (actual duration over original duration). This  

report will identify those activities which have changed and check to  

see how they compare with the mean. Files must be entered with the most  

recent file first, followed by successively older files.  

ENDTEXT  

  

SET CONFIRM ON  

STORE 1 TO numfiles  

STORE 0 TO fail  

  

DO WHILE .T.  

  STORE 0 TO fail  

  IF numfiles > 5  

    EXIT  

  UNDIF  

  @ 8,5 SAY 'INPUT THE FILENAME FOR SCHEDULE FILE '  

  IF numfiles = 1  

    filsched1 = filsched1 + ""  

  @ 8,42 SAY numfiles PICTURE '9'  

  @ 9,20 GET filsched1  

  READ  

  @ 10,20 SAY " "  

  IF filsched1 = ''  

    EXIT  

  ENDIF  

  filsched1 = UPPER(filsched1)  

  DO check WITH filsched1,fail  

  IF option = 0  

    RETURN

```

```

ENDIF
IF fail <> 1
    STORE numfiles + 1 TO numfiles
ENDIF
ENDIF
IF numfiles = 2
    filsched2 = filsched2 + "
    @ 8,42 SAY numfiles PICTURE '9'
    @ 9,20 GET filsched2
READ
@ 10,20 SAY ""
IF filsched2 = ''
    EXIT
ENDIF
filsched2 = UPPER(filsched2)
DO check WITH filsched2,fail
IF option = 0
    RETURN
ENDIF
IF fail <> 1
    IF filsched2 = filsched1
        DO disperr WITH 1,filsched2
        IF option = 0
            RETURN
        ENDIF
        STORE 1 TO fail
    ENDIF
ENDIF
IF fail <> 1
    STORE numfiles + 1 TO numfiles
ENDIF
ENDIF

IF numfiles = 3
    filsched3 = filsched3 + "
    @ 8,42 SAY numfiles PICTURE '9'
    @ 9,20 GET filsched3
READ
@ 10,20 SAY ""
IF filsched3 = ''
    EXIT
ENDIF
filsched3 = UPPER(filsched3)
DO check WITH filsched3,fail
IF option = 0
    RETURN
ENDIF
IF fail <> 1
    IF (filsched3 = filsched2) .OR. (filsched3=filsched1)
        DO disperr WITH 1,filsched3
        IF option = 0
            RETURN

```

```

        ENDIF
        STORE 1 TO fail
    ENDIF
ENDIF
IF fail <> 1
    STORE numfiles+1 TO numfiles
ENDIF
ENDIF

IF numfiles = 4
    filsched4 = filsched4 + "
    @ 8,42 SAY numfiles PICTURE '9'
    @ 9,20 GET filsched4
READ
@ 10,20 SAY "
IF filsched4 = ''
    EXIT
ENDIF
filsched4 = UPPER(filsched4)
DO check WITH filsched4,fail
IF option = 0
    RETURN
ENDIF
IF fail <> 1
    IF (filsched4=filsched3) .OR. (filsched4=filsched2) .OR. (filsched4=filsched1)
        DO disperr WITH 1,filsched4
        IF option = 0
            RETURN
        ENDIF
        STORE 1 TO fail
    ENDIF
ENDIF
IF fail <> 1
    STORE numfiles + 1 TO numfiles
ENDIF
ENDIF

IF numfiles = 5
    filsched5 = filsched5 + "
    @ 8,42 SAY numfiles PICTURE '9'
    @ 9,20 GET filsched5
READ
@ 10,20 SAY "
IF filsched5 = ''
    EXIT
ENDIF
filsched5 = UPPER(filsched5)
DO check WITH filsched5,fail
IF option = 0
    RETURN
ENDIF
IF fail <> 1

```

```

    IF (filsched5=filsched4) .OR. (filsched5=filsched3) .OR. (filsched5=filsched2) .OR.
(filsched5=filsched1)
        DO disperr WITH 1,filsched5
        IF option = 0
            RETURN
        ENDIF
        STORE 1 TO fail
    ENDIF
    IF fail <> 1
        STORE numfiles + 1 TO numfiles
    ENDIF
ENDIF
ENDDO
numfiles = numfiles - 1
IF numfiles >= 2
    RETURN
ELSE
    CLEAR
    @ 10,10
    TEXT
At least two files to run this report.

```

If you select "R" you may enter files again. If you enter a "Q", the report will be skipped. If you enter a "D", you will see a list of database files, then be given the opportunity to enter files again.

```

ENDTEXT
o_confirm = 'R'
DO WHILE .T.
    @ 19,20 SAY "Your choice (R,Q,D):" GET o_confirm
    READ
    o_confirm = UPPER(o_confirm)
    DO CASE
        CASE o_confirm = "R"
            EXIT
        CASE o_confirm = "D"
            CLEAR
            DJR
            WAIT
        CASE o_confirm = "Q"
            numfiles = 0
            RETURN
    ENDCASE
    @ 19,20 SAY ""
    o_confirm = "R"
ENDDO
ENDIF
SET CONFIRM OFF
RETURN
ENDDO

```

\*: EOF: PAR\_O.PRG  
^Z

```

*:*****  

*: Program: NTFST.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/26/89 17:04  

*: Called by: SETPARMS.PRG  

*: Calls: DISPERR (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:41 SNAP! version 3.12e  

*:*****  

* PROGRAM : NTFST.PRG  

* PURPOSE : INPUT THE FILE NAMES FOR THE NTFSAP SYSTEM

```

STORE 0 TO fail

```

PUBLIC cur,tar,curl,tarl,Line,page,rpt1,rpt2,comments
PUBLIC wsact,wstitle,wses,wsef,wslf,wsls,wsas,wsaf,wstf,wsod,wsrd,wspct
PUBLIC wasuc,wslag,wsrel
*
STORE 0 TO page,Line,pct
STORE SPACE(8) TO cur
STORE SPACE(8) TO curl
STORE SPACE(8) TO curndx
STORE SPACE(8) TO curlndx
STORE SPACE(8) TO tar
STORE SPACE(8) TO tarl
STORE SPACE(8) TO tarndx
STORE SPACE(8) TO tarlndx
STORE SPACE(10) TO wsact
STORE SPACE(10) TO wssuc
STORE SPACE(1) TO wsrel
CLEAR
< 2,20 SAY "COMPARISON REPORTS -- DATABASE SPECIFICATION"
STORE C TO fail

@ 5,5 SAY "CURRENT SCHEDULE FILE -- " + ntfsched1
@ 6,5 SAY "CURRENT LOGIC FILE -- " + ntflog1cl

`` WHILE T.
ntfsched2 = ntfsched2 + "
@ 8,5 SAY 'INPUT THE FILENAME FOR THE PREVIOUS SCHEDULE FILE '
@ 10 20 GET ntfsched2
READ
ntfsched2 = TRIM(ntfsched2)
@ 11,1 SAY "
IF ntfsched2 = ntfsched1
DO disperr WITH 1,ntfsched2

```

```

ELSE
  IF .NOT. FILE(ntfsched2)
    DO disperr WITH 3,ntfsched2
  ELSE
    EXIT
  ENDIF
ENDIF
IF option = 0
  RETURN
ENDIF
ENDDO

DO WHILE .T.
  ntflogic2 = ntflogic2 + "
  @ 13, 5 SAY 'INPUT THE FILENAME FOR THE PREVIOUS LOGIC DATABASE '
  @ 15,20 GET ntflogic2
  READ
  ntflogic2 = TRIM(ntflogic2)
  @ 16,10 SAY "
  IF (ntflogic2=ntflogic1) .OR. (ntflogic2=ntfsched2) .OR. (ntflogic2=ntfsched1)
    DO disperr WITH 1,ntflogic2
  ELSE
    IF .NOT. FILE(ntflogic2)
      DO disperr WITH 3,ntflogic2
    ELSE
      EXIT
    ENDIF
  ENDIF
  IF option = 0
    RETURN
 ENDIF
ENDDO
RETURN

*: EOF: NTFST.PRG
^Z

```

```

*:*****
*:
*:      Program: HEADER1.PRG
*:
*:      System: PROJECT MANAGEMENT TOOLS
*:      Author: ASC/RCE and USA-CERL
*:      Copyright (c) 1989, USA-CERL/ASC/RCE
*:      Last modified: 04/27/89      1:14
*:
*:      Called by: REPA.PRG
*:                  : REPB.PRG
*:                  : REPC.PRG
*:                  : REPD.PRG
*:                  : REPZ.PRG
*:                  : REPF.PRG
*:                  : REPM.PRG
*:
*:      Documented 04/27/89 at 10:41           SNAP! version 3.12e
*:*****
** HEADER1.PRG
? eop
page=page+1
?? &header_on
? SPACE(18)+'PROJECT MANAGEMENT TOOLS: REPORT '+reportch
? &header_mid
? SPACE(8)+'AREA : '+areav&area+SPACE(76)+'PAGE #' +SUBSTR(STR(page), 6, 9)
? &header_off
? SPACE(7)+' SCHEDULE'                                EARLY    EARLY
LATE      LATE      TOTAL   ORIG    REM    PCT'
? SPACE(7)+'ACTIVITY ID    TITLE (from Schedule files)'          START    FINISH
START      FINISH     FLOAT   DUR     DUR    COM'
? SPACE(7)+'-----'-----'-----'-----'-----'-----'
? -----
curline=8  && Current line
header2=.T.
RETURN
*: EOF: HEADER1.PRG
^Z

```

```

*:*****  

*: Program: HEADER3.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      1:16  

*: Called by: REPJ.PRG  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  

*:*****  

** HEADER3.PRG ~ Header for Report J  

? eop  

page=page+1  

?? &header_on  

? SPACE(17)+'PROJECT MANAGEMENT TOOLS: REPORT '+reportch  

? &header_mid  

? SPACE(44)+'OUT-OF-SEQUENCE ACTITIVITIES ON FLOOR '+F+' : '+floorv&f+SPACE(20)+'PAGE  

#'+SUBSTR(STR(page), 6, 9)  

? &header_off  

? ' SCHEDULE                                EARLY      EARLY      LATE  

    LATE      TOTAL    ORIG     REM     PCT     SORT'  

? '      ACTIVITY ID    TITLE (from Schedule files)          START      FINISH  

START      FINISH      FLOAT     DUR     DUR     COM     CODE'  

? '      -----      -----      -----      -----      -----      -----      -----  

-----      -----      -----      -----      -----      -----      -----  

?  

curline=8  && Current line  

header2=.T.  

RETURN  

*: EOF: HEADER3.PRG  

^Z

```

```
*****  
*: Program: HEADER4.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/27/89      1:17  
*: Called by: REPJ.PRG  
*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  
*****  
** HEADER4.PRG - Header for Report J  
? eop  
page=page+1  
?? &header_on  
? SPACE(17)+'PROJECT MANAGEMENT TOOLS: REPORT '+reportch  
? &header_mid  
? SPACE(20)+'THE FOLLOWING DRYWALL ACTIVITIES WERE BEGUN BEFORE THE EARLIEST ROOFING ACTIVITY WAS  
STARTED'+SPACE(10)+'PAGE #' +STR(page,6)  
? &header_off  
? '      SCHEDULE                                EARLY      EARLY      LATE  
    LATE      TOTAL      ORIG      REM      PCT      SORT'  
? '      ACTIVITY ID      TITLE (from Schedule files)                      START      FINISH  
START      FINISH      FLOAT      DUR      DUR      COM      CODE'  
? '      -----      -----      -----      ----      ---      -----      ,  
-----      -----      -----      ----      ---      -----      ,  
?  
curline=8  && Current line  
header2=.T.  
RETURN  
*: EOF: HEADER4.PRG  
^Z
```

```
*****  

*: Program: FIND1.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      2:47  

*: Called by: REPK.PRG  

*: Calls: FIND2.PRG  

*:       : FIND3.PRG  

*: Indexes: SUCC_ACT.NDX  

*:       : SUCC_SUC.NDX  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12a  

*****
```

PARAMETERS PATH,npth

```
PRIVATE msec,mflg

STORE 0 TO msec
SELECT B
SET INDEX TO succ_act
STORE 1 TO loc
STORE "" TO curr
STORE 0 TO cur_rec
DO WHILE loc <= LEN(PATH)
  DO WHILE SUBSTR(PATH,loc,1) <> "."
    STORE curr + SUBSTR(PATH,loc,1) TO curr
    STORE loc + 1 TO loc
    IF loc > LEN(PATH)
      EXIT
    ENDIF
  ENDDO
  STORE loc + 1 TO loc
  STORE VAL(curr) TO cur_rec
  IF cur_rec <> 0
    SELECT D
    GOTO cur_rec
    SELECT B
    SEEK d->act
    SELECT D
    STORE 0 TO cur_rec
    STORE "" TO curr
    DO find2
    IF curline > maxlinesk - 3
      IF "P" $ output
        EJECT
    ENDIF
  ENDIF
ENDIF
```

```

ENDIF
STORE .T. TO newpage
STORE 0 TO curline
ENDIF
ENDIF
ENDDO
?
?
?
DO find3 WITH 3

IF curline > maxlinesk - 7
DO find3 WITH 7
ENDIF
? " Activities on this Critical Path with less than "+LTRIM(STR(mpl))+ men per day"
?
? " ACTIVITY ID      MEN/DAY"
? " -----      -----"
?
DO find3 WITH 5
STORE 1 TO msec
STORE .T. TO mflg
STORE 1 TO loc
STORE "" TO curr
STORE 0 TO cur_rec
DO WHILE loc <= LEN(PATH)
DO WHILE SUBSTR(PATH,loc,1) <> "."
STORE curr + SUBSTR(PATH,loc,1) TO curr
STORE loc + 1 TO loc
IF loc > LEN(PATH),
EXIT
ENDIF
ENDIF
STORE loc + 1 TO loc
STORE VAL(curr) TO cur_rec
IF cur_rec <> 0
SELECT D
GOTO cur_rec
STORE 0 TO cur_rec
STORE "" TO curr
IF mh_day < mpl .AND. mh_day > 0
? "+act+STR(mh_day,12)
DO find3 WITH 1
STORE .F. TO mflg
ENDIF
ENDIF
ENDDO
STORE 0 TO msec
IF mflg
? "      NONE"
DO find3 WITH 1
ENDIF

```

```

?
?
?
DO find3 WITH 3

IF curline > maxlinesk - 8
  DO find3 WITH 8
ENDIF
? " Activities on this Critical Path with more than "+LTRIM(STR(durat))+ " days remaining duration"
?

? " ACTIVITY ID      REMAINING"
? "                  DURATION "
? " -----      -----"
?

DO find3 WITH 6
STORE .T. TO mflg
STORE 2 TO msec
STORE 1 TO loc
STORE "" TO curr
STORE 0 TO cur_rec
DO WHILE loc <= LEN(PATH)
  DO WHILE SUBSTR(PATH,loc,1) <> "."
    STORE curr + SUBSTR(PATH,loc,1) TO curr
    STORE loc + 1 TO loc
    IF loc > LEN(PATH)
      EXIT
    ENDIF
  ENDDO
  STORE loc + 1 TO loc
  STORE VAL(curr) TO cur_rec
  IF cur_rec <> 0
    SELECT D
    GOTO cur_rec
    STORE 0 TO cur_rec
    STORE "" TO curr
    IF rd > durat
      ? " "+act+STR(rd,13)
      DO find3 WITH 1
      STORE .F. TO mflg
    ENDIF
  ENDIF
  ENDDO
  STORE 0 TO msec
  IF mflg
    ? "      NONE"
    DO find3 WITH 1
  ENDIF
?
?
?
DO find3 WITH 3

```

```

IF curline > maxlinesk - 8
  DO find3 WITH 8
ENDIF

? " Activities on this Critical Path with relation durations greater than "+LTRIM(STR(reldur))+"
percent"

?
? " ACTIVITY ID      LAG      REMAINING      RELATION"
? "                      DURATION      DURATION"
? " -----      -----      -----"
?

DO find3 WITH 6
SELECT B
SET INDEX TO succ_suc
STORE .T. TO mflg
STORE 3 TO msec
STORE 1 TO loc
STORE "" TO curr
STORE 0 TO cur_rec
DO WHILE loc <= LEN(PATH)
  DO WHILE SUBSTR(PATH,loc,1) <> "."
    STORE curr + SUBSTR(PATH,loc,1) TO curr
    STORE loc + 1 TO loc
    IF loc > LEN(PATH)
      EXIT
    ENDIF
  ENDDO
  STORE loc + 1 TO loc
  STORE VAL(curr) TO cur_rec
  IF cur_rec <> 0
    SELECT D
    GOTO cur_rec
    SELECT B
    SEEK d->act
    STORE 0 TO cur_rec
    STORE "" TO curr
    IF d->rd > 0
      IF 100*(lag/d->rd) > reldur
        ? " "+d->act+STR(lag,7)+STR(d->rd,11)+STR(100*(lag/d->rd),11)
        DO find3 WITH 1
        STORE .F. TO mflg
      ENDIF
    ENDIF
  ENDIF
ENDDO
STORE 0 TO msec
IF mflg
  ? "      NONE"
  DO find3 WITH 1
ENDIF
?
STORE .T. TO newpage

```

RETURN  
\*: EOF: FINDI.PRG  
^Z

```

*:*****  

*: Program: STND_FND.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      0:20  

*: Called by: REPO.PRG  

*: Indexes: MRGNDX.NDX  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e  

*:*****  

STORE 0 TO sdev  

STORE .T. TO newpage  

STORE 0 TO page  

STORE 0 TO matches  

STORE 0 TO curline  

STORE 0 TO areamat  

INDEX ON SUBSTR(act,5,2) + act TO mrgndx  

GO TOP  

newpage = .T.  

  

DO WHILE .NOT. EOF()  

  STORE SUBSTR(act,5,2) TO stnd  

  STORE RECNO() TO start  

  STORE 0 TO N,mean,prod,prodsq  

  DO WHILE (SUBSTR(act,5,2) = stnd) .AND. .NOT. EOF()  

    STORE prod + ((af-as)/od) TO prod  

    STORE prodsq + (((af-as)/od)**2) TO prodsq  

    STORE N + 1 TO N  

    SKIP + numfiles  

  ENDDO  

  GO start  

  STORE prod / N TO mean  

  IF (N > 1)  

    sdev = SQRT((prodsq - ((prod**2)/N))/(N-1))  

  ELSE  

    sdev=0 0  

  ENDIF  

  IF newpage  

    ? eop  

    curline = 0  

    page = page + 1  

    ? &header_on  

    ? SPACE(17)+' SCHEDULING ANALYSIS SYSTEM: REPORT O'  

  IF SUBSTR(act,5,2)$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC'  

    ? '   ICS STANDARD CODE = '+stnd+': '+stndv&stnd+SPACE(36)+'PAGE #'&STR(page,6)  

  ELSE  

    ? '   ICS STANDARD CODE = '+stnd+': '+stnd+SPACE(36)+'PAGE #'&STR(page,6)

```



```

ENDIF
SELECT G
APPEND BLANK
REPL act WITH h->act,TITLE WITH h->title es WITH h->es
REPL esa WITH h->esa,ef WITH h->ef,efa WITH h->efa
REPL ls WITH h->ls,lsa WITH h->lsa,lf WITH h->lf,lfa WITH h->lfa
REPL as WITH h->as,af WITH h->af,tf WITH h->tf,ffl WITH h->ffl
REPL od WITH h->od,rd WITH h->rd,pct WITH h->pct,resp WITH h->resp
REPL lvs WITH h->lvs,SORT WITH h->sort,rcp WITH h->rcp
REPL rut WITH h->rut,upd WITH h->upd,svc WITH h->svc
REPL dol_day WITH h->dol_day,mh_day WITH h->mh_day
REPL changed WITH prod
SELECT H
STORE matches + 1 TO matches
STORE curline + 1 TO curline
SKIP + (numfiles - 1)
ELSE
  SELECT G
  APPEND BLANK
  REPL act WITH h->act,TITLE WITH h->title,es WITH h->es
  REPL esa WITH h->esa,ef WITH h->ef,efa WITH h->efa
  REPL ls WITH h->ls,lsa WITH h->lsa,lf WITH h->lf,lfa WITH h->lfa
  REPL as WITH h->as,af WITH h->af,tf WITH h->tf,ffl WITH h->ffl
  REPL od WITH h->od,rd WITH h->rd,pct WITH h->pct,resp WITH h->resp
  REPL lvs WITH h->lvs,SORT WITH h->sort,rcp WITH h->rcp
  REPL rut WITH h->rut,upd WITH h->upd,svc WITH h->svc
  REPL dol_day WITH h->dol_day,mh_day WITH h->mh_day
  REPL changed WITH 1
  SELECT H
ENDIF
IF curline > maxlineso - 3
  newpage=.T.
  EXIT
ENDIF
IF .NOT. EOF()
  SKIP
ENDIF
ENDDO
IF SUBSTR(act,5,2) <> stnd
?
IF stnd$'01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,DR,PM,PR,SC'
? SPACE(25)+'++++ For '+stnd+' : '+TRIM(stndv&stnd)+' STATISTICS ARE : MEAN = '
ELSE
? SPACE(25)+'++++ For '+stnd+' : '+' STATISTICS ARE : MEAN = '
ENDIF
?? LTRIM(STR(mean,9,2))+'; STANDARD DEVIATION = '+LTRIM(STR(sdev,9,2))+'+ +++'
? SPACE(25)+'There were '+LTRIM(RTRIM(STR(areamat)))+' activities which had slipped.'
newpage=.T.
ENDIF
STORE 0 TO areamat
ENDDO
SELECT G

```

```
GO TOP
DO WHILE .NOT. EOF()
    STORE .F. TO fnd
    IF changed <> 1
        STORE .T. TO fnd
        EXIT
    ENDIF
    SKIP +1
ENDDO
IF .NOT. fnd
    ZAP
ELSE
    ?
    ? 'The new schedule file is NEWSCHED.DBF. The changed field indicates the productivity rate.'
    ?
ENDIF
? eop
?
? &footer_on
? SPACE(50)+'*** END OF REPORT O ***'
? SPACE(20)+'*** THERE WERE '+LTRIM(STR(matches))+ ' ACTIVITY ID NUMBERS WHICH MATCHED THESE SEARCH
CRITERIA. ***'
? &footer_off
? eop
RETURN
*: EOF: STND_FND.PRG
^Z
```

```

*:*****  

*: Program: TITLE1.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      4:58  

*: Called by: REPR.PRG  

*:           : REPS.PRG  

*:           : REPT.PRG  

*:           : REPUPRG  

*:           : REPV.PRG  

*:           : REPW.PRG  

*:           : REPX.PRG  

*:           : PRINTACT.PRG  

*:           : PRINTPRE.PRG  

*: Calls: TOTOP          (procedure in PROCS.PRG)  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e
*:*****  

*  

* THIS PROGRAM IS TO PRINT REPORT TILE, PAGE NUMBER, DATE,  

* ACTIVITY CONTENTS FOR "SCHEDULE.PRG"  

*  

SET TALK OFF
DO totop
Line = 2
? &intro_on
? &header_on
?     &header_mid
? SPACE(40)+'NATIONAL TEST FACILITY SCHEDULE ANALYSIS PROGRAM (1.0b)'
Line = Line + 2
? &header_off

DO CASE
CASE rpt1 = '1'
    ? "THE FOLLOWING ACTIVITIES HAVE BEEN DELETED FROM THE MOST RECENT SCHEDULE"
CASE rpt1 = '2'
    ? "THE FOLLOWING ACTIVITIES HAVE BEEN ADDED TO THE MOST RECENT SCHEDULE"
CASE rpt1 = '3'
    ? "THE FOLLOWING ACTIVITIES HAVE INCONSISTENCY BETWEEN PCT AND ACTUAL START/FINISH DATES"
CASE rpt1 = '4'
    ? "THE FOLLOWING ACTIVITIES HAVE CHANGES IN DESCRIPTION BETWEEN SCHEDULES"
CASE rpt1 = '5'
    ? SPACE(44)+ 'THE FOLLOWING ACTIVITIES HAVE SIGNIFICANT CHANGES IN ORIGINAL DURATION BETWEEN
SCHEDULES'
    Line = Line + 1
    ? "(CURRENT OD INCREASED, TOTAL FLOAT LESS THAN 5, AND INCREASING RATIO > 10 % )"
CASE rpt1 = '6'

```

```

? "THE FOLLOWING ACTIVITIES HAVE CHANGES IN THE CURRENT SCHEDULE"
CASE rpt1 = '7'
    ? "THE FOLLOWING ACTIVITIES HAVE CHANGES IN RELATION TO THEIR SUCCESSORS IN THE CURRENT SCHEDULE"
ENDCASE
Line = Line + 2
?
? 'REPORT WAS RUN ON '
?? DATE()
IF rpt1 <> '3'
    ?? SPACE(43)+ "THE CURRENT FILE IS: "
    ?? UPPER(ntfsched1)
ELSE
    ?? SPACE(43)+ "IN THE CURRENT FILE: "
    ?? UPPER(ntfsched1)
ENDIF
?? SPACE(20)+ 'PAGE # '
page = page +1
?? STR(page,3)
Line = Line + 1
IF rpt1 <> '3'
    ? SPACE(70)+"THE PREVIOUS FILE IS: "
    ?? UPPER(ntfsched2)
ENDIF
Line = Line +2
?
IF rpt1 <> '7'
    ? 'T '
    ?? SPACE(57)+ 'EARLY      EARLY      LATE      LATE      ACTUAL      ACTUAL      TOTAL ORIG REM PCT
COMMENTS'
    Line = Line + 1
    ? 'L NUMBER DESCRIPTION'
    ?? SPACE(38)+ 'START      FINISH      START      FINISH      START      FINISH      FLOAT DUR DUR COM'
ELSE
    ? 'T ACT      ACTIVITY'+SPACE(49)
    ?? 'ACTUAL      ACTUAL      TOTAL ORIG REM PCT ACTIVITY      ACTIVITY'+SPACE(39)
    ?? 'REL LAG COMMENT'
    Line = Line + 1
    ? 'L NUMBER DESCRIPTION'+SPACE(46)
    ?? 'START      FINISH      FLOAT DUR DUR COM NUMBER      DESCRIPTION'
ENDIF
?
*
Line = Line + 2
*: EOF: TITLE1.PRG
^Z

```

```

*:*****  

*:  

*:      Program: PRINTACT.PRG  

*:  

*:      System: PROJECT MANAGEMENT TOOLS  

*:      Author: ASC/RCE and USA-CERL  

*:      Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/05/89      9:16  

*:  

*:      Called by: REPR.PRG  

*:          : REPS.PRG  

*:          : REPT.PRG  

*:          : REPUS.PRG  

*:          : REPV.PRG  

*:          : REPW.PRG  

*:  

*:      Calls: TITLE1.PRG  

*:  

*: Documented 04/27/89 at 10:41           SNAP! version 3.12e
*:*****  

*:  

* SUBROUTINE TO PRINT ONE ACTIVITY  

*  

Line = Line + 1
IF Line > 56
    DO title1
    Line = Line + 1
ENDIF
? act
?? ''
?? TITLE
?? ''
?? es
?? ''
?? ef
?? ''
?? ls
?? ''
?? lf
?? ''
?? as
?? ''
?? af
?? ''
?? tf
?? ''
?? od
?? ''
?? rd
?? ''
?? pct
*

```

```
IF rpt2 = '1'  
?? ??  
?? comments  
ENDIF  
*: EOF: PRINTACT.PRG  
^Z
```

```
*:*****  
*: Program: PRINTPRE.PRG  
*: System: PROJECT MANAGEMENT TOOLS  
*: Author: ASC/RCE and USA-CERL  
*: Copyright (c) 1989, USA-CERL/ASC/RCE  
*: Last modified: 04/27/89 2:36  
*: Called by: REPX.PRG  
*: Calls: TITLE1.PRG  
*: Documented 04/27/89 at 10:42 SNAP! version 3.12e  
*:  
*  
* SUBROUTINE TO PRINT ONE PREDECESSOR ACTIVITY  
*  
Line = Line + 1  
IF Line > 56  
  DO title1  
  Line = Line + 1  
ENDIF  
? act+" "+TITLE+SPACE(9)+DTOC(as)+SPACE(2)+DTOC(af)+STR(tf,8)+STR(od,6)  
?? STR(rd,6)+STR(pct,6,2)  
*  
*: EOF: PRINTPRE.PRG  
^Z
```

```
*****  

*: Program: FIND2.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      2:48  

*: Called by: FIND1.PRG  

*: Documented 04/27/89 at 10:42           SNAP! version 3.12e  

*****
```

```
IF newpage
  curline = 6
  page = page + 1
? eop
? &header_on
? SPACE(17)+"SCHEDULING ANALYSIS SYSTEM: REPORT K"
?
? " CRITICAL PATH #"+STR(np,3)+SPACE(36)+"PAGE #"+STR(page,3)
? &header_off
? " ACTIVITY      TITLE          EARLY      EARLY      "
??" LATE        LATE      ACTUAL      ACTUAL      TOTAL      REM      PCT      ORIG      LAG      REL "
? " ID          START      FINISH      START      FINISH      FLOAT      DUR      COM      DUR      "
? " -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      "
??" -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      -----      "
?
newpage = .F.
ENDIF
? " "+act+"    "+TITLE+"    "+DTOC(es)+"    "+DTOC(ef)+"    "+DTOC(ls)
?? "    "+DTOC(lf)+"    "+DTOC(as)+"    "+DTOC(af)+STR(tf,7)
?? STR(rd,7)+STR(pct,6)+STR(od,6)+STR(b->lag,6)+"    "+b->rel
curline = curline + 1
IF curline >= maxlinesk
  STORE .T. TO newpage
ENDIF

RETURN
*: EOF: FIND2.PRG
^Z
```

```

*:*****  

*: Program: FIND3.PRG  

*: System: PROJECT MANAGEMENT TOOLS  

*: Author: ASC/RCE and USA-CERL  

*: Copyright (c) 1989, USA-CERL/ASC/RCE  

*: Last modified: 04/27/89      3:43  

*: Called by: FIND1.PRG  

*: Documented 04/27/89 at 10:42           SNAP! version 3.12e
*:*****
```

PARAMETERS mline

```

STORE curline + mline TO curline
IF curline > maxlinesk
  page = page + 1
  ? eop
  ? &header_on
  ? SPACE(17)+"SCHEDULING ANALYSIS SYSTEM: REPORT K"
  ?
  ? " Critical Path #"+STR(npah,3)+SPACE(36)+"PAGE #"+STR(page,3)
  ? &header_off
  ?

STORE 3 TO curline
DO CASE
CASE msec > 1
  ? " ACTIVITY ID      MEN/DAY"
  ? " -----      -----"
  ?
STORE 3 TO curline
CASE msec = 2
  ? " ACTIVITY ID      REMAINING"
  ? "                   DURATION "
  ? " -----      -----"
  ?
STORE 4 TO curline
CASE msec = 3
  ? " ACTIVITY ID      LAG      REMAINING      RELATION"
  ? "                   DURATION      DURATION"
  ? " -----      -----      -----"
  ?
STORE 4 TO curline
ENDCASE
newpage = .F.
ENDIF

RETURN
*: EOF: FIND3.PRG
^Z
```

## **USAEC ERL DISTRIBUTION**

Chief of Engineers  
ATTN: CEHEC-IM-LP (2)  
ATTN: CEHEC-IM-LH (2)  
ATTN: CECC-P  
ATTN: CECW  
ATTN: CERD-L  
ATTN: CEMP-C  
ATTN: CEMP-CP  
ATTN: CEMP-CM  
ATTN: CEMP-CE

US Army Engineer Districts  
ATTN: C/Const. Div. (41)

US Army Engr Divisions  
ATTN: C/Const. Div (13)

General Services Administration  
St. Louis, MO 63120  
Kansas City, MO 64131

DLA ATTN: DLA-WI 22304

DNA ATTN: NADS 20305

Fort Belvoir, VA 22060  
ATTN: CECC-R

NAVFAC  
ATTN: Division Offices (11)  
ATTN: Facilities Engr Cmd (9)  
ATTN: Naval Constr Battalion Ctr

Omaha District  
ATTN: CEMRO-CD 68102

Defense Technical Info. Center 22304  
ATTN: DTIC-FAB (2)